In [1]:
```
import os
import sys
import gc
import cv2
import random
import shutil
import numpy as np
from PIL import Image
from random import sample
from pyunpack import Archive
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from scipy.ndimage import map_coordinates
```

In [2]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.nn.modules.utils as nn_utils
from torchvision.transforms import PILToTensor
from typing import Any, Callable, Dict, List, Optional, Union, Tuple
from diffusers.models.unet_2d_condition import UNet2DConditionModel
from diffusers import DDIMScheduler
from diffusers import StableDiffusionPipeline
```

```
2024-04-27 01:46:45.262999: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in performa
nce-critical operations.
To enable the following instructions: SSE4.1 SSE4.2 AVX AVX2 FMA, in other operat
ions, rebuild TensorFlow with the appropriate compiler flags.
```

In [3]:
```
#Archive('FIRE.7z').extractall(os.getcwd())
```

In [4]:
```
#shutil.rmtree(os.path.join(os.getcwd(),'FIRE_Image_Registration_Results'))
```

In [5]:
```
path = os.path.join(os.getcwd(),'FIRE_Image_Registration_Results/Stage1')
os.makedirs(path,exist_ok=True)
```

In [6]:
```
path = os.path.join(os.getcwd(),'FIRE_Image_Registration_Results/Stage2')
os.makedirs(path,exist_ok=True)
```

In [7]:
```
path = os.path.join(os.getcwd(),'FIRE_Image_Registration_Results/Final_Registrat
os.makedirs(path,exist_ok=True)
```

Note:

Some of the code cells were referenced from the paper titled "Emergent Correspondence from Image Diffusion." Please cite their paper as follows:

```
@inproceedings{tang2023emergent,
  title={Emergent Correspondence from Image Diffusion},
  author={Luming Tang and Menglin Jia and Qianqian Wang and Cheng Perng
Phoo and Bharath Hariharan},
  booktitle={Thirty-seventh Conference on Neural Information Processing
Systems},
  year={2023},
```

```
  url={https://openreview.net/forum?id=ypOiXjdfnU}
}
```

In [3]:

```
class MyUNet2DConditionModel(UNet2DConditionModel):
    """
    Customized 2D U-Net conditioned model inherited from `UNet2DConditionModel`.

    This model extends the original `UNet2DConditionModel` to incorporate additi
    such as encoder hidden states, attention mask, and cross-attention keyword a
    """
    def forward(
        self,
        sample: torch.FloatTensor,
        timestep: Union[torch.Tensor, float, int],
        up_ft_indices,
        encoder_hidden_states: torch.Tensor,
        class_labels: Optional[torch.Tensor] = None,
        timestep_cond: Optional[torch.Tensor] = None,
        attention_mask: Optional[torch.Tensor] = None,
        cross_attention_kwargs: Optional[Dict[str, Any]] = None):
        """
        Forward method for `MyUNet2DConditionModel`.

        Args:
            sample (torch.FloatTensor): Noisy inputs tensor with shape (batch, c
            timestep (torch.FloatTensor or float or int): Timesteps for each bat
            up_ft_indices (list): List of upsampling indices.
            encoder_hidden_states (torch.FloatTensor): Encoder hidden states wit
            class_labels (Optional[torch.Tensor], default=None): Class labels te
            timestep_cond (Optional[torch.Tensor], default=None): Timestep condi
            attention_mask (Optional[torch.Tensor], default=None): Mask to avoid
            cross_attention_kwargs (Optional[dict], default=None): Keyword argum

        Returns:
            dict: Dictionary containing upsampled features (`up_ft`).
        """

        # By default samples have to be AT least a multiple of the overall upsam
        # The overall upsampling factor is equal to 2 ** (# num of upsampling la
        # However, the upsampling interpolation output size can be forced to fit
        # on the fly if necessary.
        default_overall_up_factor = 2**self.num_upsamplers

        # upsample size should be forwarded when sample is not a multiple of `de
        forward_upsample_size = False
        upsample_size = None

        if any(s % default_overall_up_factor != 0 for s in sample.shape[-2:]):
            # logger.info("Forward upsample size to force interpolation output s
            forward_upsample_size = True

        # prepare attention_mask
        if attention_mask is not None:
            attention_mask = (1 - attention_mask.to(sample.dtype)) * -10000.0
            attention_mask = attention_mask.unsqueeze(1)

        # 0. center input if necessary
        if self.config.center_input_sample:
            sample = 2 * sample - 1.0
```

```
        # 1. time
        timesteps = timestep
        if not torch.is_tensor(timesteps):
            # TODO: this requires sync between CPU and GPU. So try to pass times
            # This would be a good case for the `match` statement (Python 3.10+)
            is_mps = sample.device.type == "mps"
            if isinstance(timestep, float):
                dtype = torch.float32 if is_mps else torch.float64
            else:
                dtype = torch.int32 if is_mps else torch.int64
            timesteps = torch.tensor([timesteps], dtype=dtype, device=sample.dev
        elif len(timesteps.shape) == 0:
            timesteps = timesteps[None].to(sample.device)

        # broadcast to batch dimension in a way that's compatible with ONNX/Core
        timesteps = timesteps.expand(sample.shape[0])

        t_emb = self.time_proj(timesteps)

        # timesteps does not contain any weights and will always return f32 tens
        # but time_embedding might actually be running in fp16. so we need to ca
        # there might be better ways to encapsulate this.
        t_emb = t_emb.to(dtype=self.dtype)

        emb = self.time_embedding(t_emb, timestep_cond)

        if self.class_embedding is not None:
            if class_labels is None:
                raise ValueError("class_labels should be provided when num_class

            if self.config.class_embed_type == "timestep":
                class_labels = self.time_proj(class_labels)

            class_emb = self.class_embedding(class_labels).to(dtype=self.dtype)
            emb = emb + class_emb

        # 2. pre-process
        sample = self.conv_in(sample)

        # 3. down
        down_block_res_samples = (sample,)
        for downsample_block in self.down_blocks:
            if hasattr(downsample_block, "has_cross_attention") and downsample_b
                sample, res_samples = downsample_block(
                    hidden_states=sample,
                    temb=emb,
                    encoder_hidden_states=encoder_hidden_states,
                    attention_mask=attention_mask,
                    cross_attention_kwargs=cross_attention_kwargs,
                )
            else:
                sample, res_samples = downsample_block(hidden_states=sample, tem

            down_block_res_samples += res_samples

        # 4. mid
        if self.mid_block is not None:
            sample = self.mid_block(
                sample,
```

```
                emb,
                encoder_hidden_states=encoder_hidden_states,
                attention_mask=attention_mask,
                cross_attention_kwargs=cross_attention_kwargs,
            )

        # 5. up
        up_ft = {}
        for i, upsample_block in enumerate(self.up_blocks):

            if i > np.max(up_ft_indices):
                break

            is_final_block = i == len(self.up_blocks) - 1

            res_samples = down_block_res_samples[-len(upsample_block.resnets) :]
            down_block_res_samples = down_block_res_samples[: -len(upsample_bloc

            # if we have not reached the final block and need to forward the
            # upsample size, we do it here
            if not is_final_block and forward_upsample_size:
                upsample_size = down_block_res_samples[-1].shape[2:]

            if hasattr(upsample_block, "has_cross_attention") and upsample_block
                sample = upsample_block(
                    hidden_states=sample,
                    temb=emb,
                    res_hidden_states_tuple=res_samples,
                    encoder_hidden_states=encoder_hidden_states,
                    cross_attention_kwargs=cross_attention_kwargs,
                    upsample_size=upsample_size,
                    attention_mask=attention_mask,
                )
            else:
                sample = upsample_block(
                    hidden_states=sample, temb=emb, res_hidden_states_tuple=res_
                )

            if i in up_ft_indices:
                up_ft[i] = sample.detach()

        output = {}
        output['up_ft'] = up_ft
        return output

class OneStepSDPipeline(StableDiffusionPipeline):
    """
    One-step Stable Diffusion Pipeline.

    Provides a one-step stable diffusion process, integrating the VAE encoding a
    """
    @torch.no_grad()
    def __call__(
        self,
        img_tensor,
        t,
        up_ft_indices,
        negative_prompt: Optional[Union[str, List[str]]] = None,
        generator: Optional[Union[torch.Generator, List[torch.Generator]]] = Non
        prompt_embeds: Optional[torch.FloatTensor] = None,
```

```
        callback: Optional[Callable[[int, int, torch.FloatTensor], None]] = None
        callback_steps: int = 1,
        cross_attention_kwargs: Optional[Dict[str, Any]] = None
    ):

        """
        Call method for `OneStepSDPipeline`.

        Args:
            img_tensor (torch.Tensor): Image tensor.
            t (torch.Tensor or int): Timesteps tensor.
            up_ft_indices (list): List of upsampling indices.
            negative_prompt (Optional[str or list], default=None): Negative prom
            generator (Optional[torch.Generator or list], default=None): Torch g
            prompt_embeds (Optional[torch.FloatTensor], default=None): Precomput
            callback (Optional[Callable], default=None): Callback function invok
            callback_steps (int, default=1): Frequency of invoking the callback.
            cross_attention_kwargs (Optional[dict], default=None): Keyword argum

        Returns:
            dict: Dictionary containing output from U-Net.
        """
        device = self._execution_device
        latents = self.vae.encode(img_tensor).latent_dist.sample() * self.vae.cc
        t = torch.tensor(t, dtype=torch.long, device=device)
        noise = torch.randn_like(latents).to(device)
        latents_noisy = self.scheduler.add_noise(latents, noise, t)
        unet_output = self.unet(latents_noisy,
                                t,
                                up_ft_indices,
                                encoder_hidden_states=prompt_embeds,
                                cross_attention_kwargs=cross_attention_kwargs)
        return unet_output


class SDFeaturizer:
    """
    Stable Diffusion Featurizer.

    Provides a mechanism to compute stable diffusion based features from an inpu
    """
    def __init__(self, sd_id='stabilityai/stable-diffusion-2-1'):
        """
        Initializes `SDFeaturizer` with a given stable diffusion model ID.

        Args:
            sd_id (str, default='stabilityai/stable-diffusion-2-1'): Stable diff
        """
        unet = MyUNet2DConditionModel.from_pretrained(sd_id, subfolder="unet")
        onestep_pipe = OneStepSDPipeline.from_pretrained(sd_id, unet=unet, safet
        onestep_pipe.vae.decoder = None
        onestep_pipe.scheduler = DDIMScheduler.from_pretrained(sd_id, subfolder=
        gc.collect()
        onestep_pipe = onestep_pipe.to("cuda")
        onestep_pipe.enable_attention_slicing()
        onestep_pipe.enable_xformers_memory_efficient_attention()
        self.pipe = onestep_pipe

    @torch.no_grad()
    def forward(self,
```

```
                img_tensor, # single image, [1,c,h,w]
                t,
                up_ft_index,
                prompt,
                ensemble_size=8):
        """
        Forward method for `SDFeaturizer`.

        Args:
            img_tensor (torch.Tensor): Single input image tensor with shape [1,
            t (torch.Tensor or int): Timesteps tensor.
            up_ft_index (int): Index for upsampling.
            prompt (str): Textual prompt for conditioning.
            ensemble_size (int, default=8): Size of the ensemble for feature ave

        Returns:
            torch.Tensor: Stable diffusion based features with shape [1, c, h, w
        """
        img_tensor = img_tensor.repeat(ensemble_size, 1, 1, 1).cuda() # ensem, c
        prompt_embeds = self.pipe._encode_prompt(
            prompt=prompt,
            device='cuda',
            num_images_per_prompt=1,
            do_classifier_free_guidance=False) # [1, 77, dim]
        prompt_embeds = prompt_embeds.repeat(ensemble_size, 1, 1)
        unet_ft_all = self.pipe(
            img_tensor=img_tensor,
            t=t,
            up_ft_indices=[up_ft_index],
            prompt_embeds=prompt_embeds)
        unet_ft = unet_ft_all['up_ft'][up_ft_index] # ensem, c, h, w
        unet_ft = unet_ft.mean(0, keepdim=True) # 1,c,h,w
        return unet_ft
```

In [9]:

```
class DFT:
    """
    RetinaRegNet (RetinaRegNetwork) utilizes DFT (Diffusion Features) for identi
    and locations between images.
    """
    def __init__(self, imgs,img_size,pts):
        """
        Initialize the DFT object.

        Parameters:
        - imgs (list): List of input image tensors.
        - img_size (int): Expected size of the image for processing.
        - pts (list): List of point tuples specifying coordinates.
        """
        self.pts = pts
        self.imgs = imgs
        self.num_imgs = len(imgs)
        self.img_size = img_size

    def unravel_index(self,index, shape):
        """
        Converts a flat index into a tuple of coordinate indices in a tensor of

        This function mimics numpy's `unravel_index` functionality, which is use
        into a tuple of coordinate indices for an array of given shape. This is
        multi-dimensional indices of a position in a flattened array.
```

```
        Parameters:
        - index (int): The flat index into the array.
        - shape (tuple of ints): The shape of the array from which the index is

        Returns:
        - tuple of ints: A tuple representing the coordinates of the index in an

        Note:
            This function operates under the assumption that indexing starts fro
        """
        out = []
        for dim in reversed(shape):
            out.append(index % dim)
            index = index // dim
        return tuple(reversed(out))

    def compute_pooled_and_combining_feature_maps(self,feature_map, hierarchy_ra
        """
        Compute pooled and stacked feature maps.

        Parameters:
        - feature_map (torch.Tensor): Input feature map.
        - hierarchy_range (int, optional): Depth of hierarchical pooling. Defaul
        - stride (int, optional): Stride for pooling. Defaults to 1.

        Returns:
        - torch.Tensor: Pooled and stacked feature map.
        """
        # List to store the pooled feature maps
        pooled_feature_maps = feature_map
        # Loop through the specified hierarchy range
        for hierarchy in range(1,hierarchy_range):
            # Average pooling with kernel size 3^k x 3^k
            win_size = 3 ** hierarchy
            avg_pool = torch.nn.AvgPool2d(win_size, stride=1, padding=win_size /
            pooled_map = avg_pool(feature_map)
            # Append the pooled feature map to the list
            pooled_feature_maps=+pooled_map
        return pooled_feature_maps

    def compute_batched_2d_correlation_maps(self, pts_list, feature_map1, featur
        """
        Computes 2D correlation maps between selected points in one feature map

        This method takes two feature maps and a list of points. It extracts fea
        at specified points, normalizes them, and then computes a batched 2D cor
        The output is a set of correlation maps, each corresponding to a point i
        feature vector correlates across the spatial dimensions of the second fe

        Parameters:
        - pts_list (list of tuples): List of points (y, x) for which the correla
        - feature_map1 (torch.Tensor): The first feature map tensor of shape (1,
        - feature_map2 (torch.Tensor): The second feature map tensor of shape (1
                                       and H2, W2 do not necessarily need to be eq

        Returns:
        - torch.Tensor: A tensor of shape (NumPoints, H2, W2) where each slice c
                        for each point in `pts_list`.
        Notes:
```

```
        The function assumes that the first dimension of feature_map1 and fe
        This method uses batch matrix multiplication and vector normalizatio
        Running this method on a GPU is recommended due to its computational
    """
    # Convert the input tensors to float16
    feature_map1 = feature_map1.to(dtype=torch.float16)
    feature_map2 = feature_map2.to(dtype=torch.float16)
    _, C, H, W = feature_map2.shape

    # Flatten feature_map2 for batch matrix multiplication
    feature_map2_flat = feature_map2.view(C, H*W)

    # Prepare a batch of point features
    points_indices = torch.tensor(pts_list)
    point_features = feature_map1[0, :, points_indices[:, 0], points_indices

    # Normalize the point features and feature_map2_flat
    point_features_norm = torch.norm(point_features, dim=1, keepdim=True)
    normalized_point_features = point_features / point_features_norm

    feature_map2_norm = torch.norm(feature_map2_flat, dim=0, keepdim=True)
    normalized_feature_map2 = feature_map2_flat / feature_map2_norm

    # Compute the correlation map for each point
    correlation_maps = torch.mm(normalized_point_features, normalized_featur

    # Reshape the correlation maps to the desired output shape (NumPoints, H
    correlation_maps = correlation_maps.view(-1, H, W)

    # Cleanup if needed
    torch.cuda.empty_cache()

    return correlation_maps

def compute_correlation_map_max_locations(self, pts_list, feature_map1, feat
    """
    Compute the maximum locations in the batched correlation maps between tw

    Parameters:
    - pts_list (list of tuples): List of points for which the correlation ma
    - feature_map1, feature_map2 (torch.Tensor): The input feature maps.

    Returns:
    - torch.Tensor: Tensor of maximum locations for each point.
    - torch.Tensor: Tensor of maximum values for each point.
    """
    enhanced_feature_map1 = self.compute_pooled_and_combining_feature_maps(f
    enhanced_feature_map2 = self.compute_pooled_and_combining_feature_maps(f
    # Compute the batched correlation maps
    batched_correlation_maps = self.compute_batched_2d_correlation_maps(pts_

    M,H2, W2 = batched_correlation_maps.shape
    #print(batched_correlation_maps.shape)

    # Find the maximum values and their locations along the last two dimensi
    max_values, max_indices_flat = torch.max(batched_correlation_maps.view(1

    x, y = zip(*[self.unravel_index(idx.item(), (H2, W2)) for idx in max_ind
    x = torch.tensor(x, device = 'cuda').view(M)
    y = torch.tensor(y, device = 'cuda').view(M)
```

```
        # Stack the coordinates to get a 2xHxW tensor
        max_locations = torch.stack((x, y)).t()

        return max_locations, max_values

    def feature_upsampling(self,ft):
        """
        Upsample the feature to match the specified image size.

        Parameters:
        - ft (torch.Tensor): Feature tensor to be upsampled.

        Returns:
        - tuple: Upsampled source and target feature maps.
        """
        with torch.no_grad():
            num_channel = ft.size(1)
            src_ft = ft[0].unsqueeze(0)
            src_ft = nn.Upsample(size=(self.img_size, self.img_size), mode='bili
            gc.collect()
            torch.cuda.empty_cache()
            trg_ft = nn.Upsample(size=(self.img_size, self.img_size), mode='bili
        return src_ft,trg_ft

    def feature_maps(self,feature_map1,feature_map2,iccl):
        """
        Processes feature maps to extract points that meet the inverse consisten

        This method computes the maximum locations of correlation between featur
        checks for inverse consistency between the mapped points. It filters the
        specified inverse consistency criteria limit (iccl), keeping only those
        distance between the original point and its double-mapped location is wi

        Parameters:
        - feature_map1 (torch.Tensor): The first feature map, used as the base f
        - feature_map2 (torch.Tensor): The second feature map, used for reverse
        - iccl (float): The maximum allowed distance (inverse consistency criter
                        its double-mapped location to be considered consistent.

        Returns:
        tuple of (list, list, list):
        - pnts (list of tuples): The points from the original feature map that m
        - rmaxs (list of floats): The maximum correlation values at these points
        - rspts (list of tuples): The corresponding points in the second feature
                                  with the points in `pnts`.
        """
        pnts,rmaxs,rspts=[],[],[]
        pts = [(int(y), int(x)) for x, y in self.pts]
        max_indices_ST, max_values_ST = self.compute_correlation_map_max_locatio
        x_prime_y_prime = max_indices_ST
        max_indices_TS, max_values_TS = self.compute_correlation_map_max_locatio
        x_prime_prime_y_prime_prime = max_indices_TS
        for i, (pt, max_idx) in enumerate(zip(self.pts, x_prime_prime_y_prime_pr
            # Calculate the distance between the point and the max correlation i
            if np.sqrt((pt[1] - max_idx.cpu()[0]) ** 2 + (pt[0] - max_idx.cpu()[
                pnts.append((int(pt[0]), int(pt[1])))
                rmaxs.append(max_values_ST[i].cpu().item())  # Assuming max_valu
                rspts.append((x_prime_y_prime[i][1].cpu().item(), x_prime_y_prim
        return pnts, rmaxs, rspts
```

In [10]:
```python
def compute_boundary(image, mean_intensity):
    """
    Compute the boundary of an image based on its mean intensity.

    Parameters:
    - image (numpy.array): The input grayscale image.
    - mean_intensity (float): Average intensity of the image to define boundarie

    Returns:
    - tuple: upper, lower, left, and right boundaries of the image region with i
    """
    # Compute the upper, lower, left, and right boundary
    upper_boundary = next((i for i, row in enumerate(image) if np.mean(row) > me
    lower_boundary = next((i for i, row in enumerate(image[::-1]) if np.mean(row

    left_boundary = next((i for i, col in enumerate(image.T) if np.mean(col) > m
    right_boundary = next((i for i, col in enumerate(image.T[::-1]) if np.mean(c

    return upper_boundary, image.shape[0]-lower_boundary, left_boundary, image.s

def is_within_boundary(kp, boundaries):
    """
    Check if a keypoint is within the specified boundaries.

    Parameters:
    - kp (cv2.KeyPoint): The keypoint to check.
    - boundaries (tuple): Tuple of (upper, lower, left, right) boundaries.

    Returns:
    - bool: True if the keypoint is within the boundaries, False otherwise.
    """
    upper, lower, left, right = boundaries
    return left <= kp.pt[0] <= right and upper <= kp.pt[1] <= lower

def SIFT_top_n_keypoints(image_path, N=250, img_shape=256, max_dist=25):
    """
    Detect top N keypoints in the given image using SIFT, considering constraint

    Parameters:
    - image_path (str): Path to the input image.
    - N (int): Number of keypoints to select. Defaults to 250.
    - img_shape (int): The size to which the image should be resized. Defaults t
    - max_dist (int): Minimum distance between selected keypoints. Defaults to 2

    Returns:
    - list: List of keypoints' positions in the form (x, y).
    """
    # Load image
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    image = cv2.resize(image, (img_shape, img_shape))

    # Initialize SIFT detector
    sift = cv2.SIFT_create()

    # Detect keypoints and compute descriptors
    keypoints, descriptors = sift.detectAndCompute(image, None)

    # Sort keypoints based on response (strength of the keypoint)
    keypoints = sorted(keypoints, key=lambda x: -x.response)
```

```
    # Determine the intensity threshold
    mean_intensity = np.mean(image)
    boundaries = compute_boundary(image, mean_intensity)

    # Select top N keypoints
    selected_keypoints = []
    for keypoint in keypoints:
        # Check if the keypoint is within the boundary
        if is_within_boundary(keypoint, boundaries):
            # Check if the pixel intensity at the keypoint is greater than the t
            if image[int(keypoint.pt[1]), int(keypoint.pt[0])] > mean_intensity:
                # Check if the keypoint is far from existing selected keypoints
                if all(cv2.norm(np.array(keypoint.pt) - np.array(kp.pt)) > max_d
                    selected_keypoints.append(keypoint)

            # Break if N keypoints are selected
            if len(selected_keypoints) == N:
                break

    return [kp.pt for kp in selected_keypoints]

def select_random_points(img, num_points=100, img_size=1200,offset=0.01,window_s
    """
    Selects a specified number of random points from an image, ensuring that eac
    meeting a defined intensity threshold within the image. The image is resized
    are chosen randomly, with each potential point undergoing validation against

    Parameters:
    - img (str): Path to the image file.
    - num_points (int, optional): The number of random points to select. Default
    - img_size (int, optional): The size to which the image is resized (assumed
    - offset (float, optional): Proportional offset to exclude points near the e
                                the image dimensions. Defaults to 0.01.
    - window_size (int, optional): Size of the square window used to check pixel
                                   Defaults to 51.
    - max_attempts_per_point (int, optional): The maximum number of attempts all
                                              that meets the criteria. Defaults to

    Returns:
    - list of tuples: A list where each tuple represents the (y, x) coordinates

    Notes:
        The function converts the image to grayscale and resizes it to img_size
        points near the image boundary by applying a boundary offset calculated
        Each point must be centered in a window (defined by 'window_size') where
        greater than or equal to 5. If the function fails to find a suitable poi
        for any location, it stops and returns the points found up to that momen
    """

    image = cv2.resize(cv2.imread(img, cv2.IMREAD_GRAYSCALE), (img_size, img_siz
    h, w = image.shape
    boundary_offset = int(offset * h)
    pts = []
    window_offset = window_size // 2  # Calculate the offset from the center of

    while len(pts) < num_points:
        attempts = 0
        while attempts < max_attempts_per_point:
            x = random.randint(boundary_offset + window_offset, h - boundary_off
```

```
            y = random.randint(boundary_offset + window_offset, w - boundary_off

            # Define the window boundaries
            x_lower = x - window_offset
            x_upper = x + window_offset + 1
            y_lower = y - window_offset
            y_upper = y + window_offset + 1

            # Check that no pixel in the window has an intensity less than 10
            if np.all(image[x_lower:x_upper, y_lower:y_upper] >= 5):
                pts.append((y, x))
                break  # Successfully found a point, break the inner loop
            attempts += 1  # Increment attempts

        if attempts == max_attempts_per_point:
            print("Maximum attempts reached, unable to find sufficient points wi
            break  # Break outer loop if max attempts is reached without finding

    return pts
```

In [11]:

```
def CLAHE_plot_cond(image,disp_clip):
    """
    Conditionally applies CLAHE to an image based on the provided clipping limit

    Parameters:
        image (np.array): The input image as a NumPy array, typically grayscale.
        disp_clip (float or str): The clip limit for CLAHE. If it is '0.0' (as a

    Returns:
        np.array: The image after applying CLAHE if `disp_clip` is not '0.0'; ot
    """
    if float(disp_clip)!=0.0:
        image = clahe(image,float(disp_clip))
    return image

def outliers_plot_condition(landmark_errors,cond):
    """
    Filters out specific outlier values from a list of landmark errors based on

    This function examines each error in the list of landmark errors and removes
    in this case, the value 10000, if the condition specified by the 'cond' para
    is False, the list is returned unchanged. This functionality can be useful f
    data before further analysis or visualization.

    Parameters:
    - landmark_errors (list of int or float): A list containing numerical values
                                              in landmarks detection.
    - cond (bool): A condition that determines whether the filtering of outliers
                 True, outliers are removed; if False, the list is returned as i

    Returns:
    - list of int or float: A list of landmark errors with specified outliers re
    """
    if cond:
        landmark_errors =[x for x in landmark_errors if x!=10000]
    return landmark_errors

def clahe(imag, clip):
    """
    Apply Contrast Limited Adaptive Histogram Equalization (CLAHE) to an image.
```

```
    This function converts an image to grayscale, applies CLAHE to enhance the i
    and then converts it back to RGB. It uses OpenCV for the CLAHE operation and
    conversions.

    Parameters:
    - imag (np.array): The input image array. Expected to be in format suitable
    - clip (float): The clipping limit for the CLAHE algorithm, which controls t
                  Higher values increase contrast.

    Returns:
    - np.array: The contrast-enhanced image in RGB format.

    Notes:
          The tile grid size for CLAHE is set to (8, 8). Adjustments to this par
          the granularity of the histogram equalization.
    """
    clahe = cv2.createCLAHE(clipLimit=clip, tileGridSize=(8, 8))
    imag = Image.fromarray(np.uint8(imag))
    imag = imag.convert('L')
    img = np.asarray(imag)
    image_equalized = clahe.apply(img)
    image_equalized_img = Image.fromarray(np.uint8(image_equalized))
    image_equalized = image_equalized_img.convert('RGB')
    image_equalized = np.asarray(image_equalized)
    return image_equalized

def compute_plot_FIRE_AUC(landmark_errors):
    """
    Function to compute and plot the success rate curve and calculate the AUC fo

    Parameters:
    - landmark_errors: List of landmark errors including outliers.
    """
    landmark_errors_sorted = sorted(landmark_errors)
    # Initialize lists for thresholds and success rates
    thresholds = list(range(26)) # 0 to 100
    success_rates = []
    # Calculate success rate for each threshold
    for threshold in thresholds:
        successful_count = sum([1 for error in landmark_errors_sorted if error <
        success_rate = successful_count / len(landmark_errors_sorted)
        success_rates.append(success_rate * 100) # convert to percentage

    # Plot the curve
    plt.plot(thresholds, success_rates, label="Success Rate Curve")
    plt.xlabel("Threshold")
    plt.ylabel("Success Rate (%)")
    plt.title("Success Rate vs. Threshold")
    plt.legend()
    plt.grid(True)
    plt.show()
    # Compute AUC
    auc = np.sum(success_rates)/ 2500 # normalize to 0-1
    print("AUC:", auc)

def plot_landmark_errors(landmark_errors,rpth,chr='All',disable_outliers=False):
    """
    Plots a graph of landmark errors over successive iterations to provide a vis
    across samples. This function is designed to help in the assessment of regis
```

```
    or computer vision tasks by plotting each landmark error against its iterati
    displays the average landmark error across all iterations.

    Parameters:
    - landmark_errors (list of float): A list containing numerical errors for ea
                                       Outliers (e.g., errors set to 10000) are au
    - rpth (str): Path where the resulting plot image will be saved.
    - chr (str, optional): Characteristic or description to include in the plot
                           used. Defaults to 'All'.
    - disable_outliers (bool, optional): If set to True, disables the automatic
                                         data. Defaults to False.

    Returns:
    - None: This function does not return any value but saves the plot to the sp

    Notes:
        This plot is useful for tracking improvements or deteriorations in landm
        It automatically filters out error values set to 10000, considering them
        is set to True.
        The function saves the plot in the directory specified by `rpth` and nam
    """
    landmark_errors=outliers_plot_condition(landmark_errors,disable_outliers)
    samples = list(range(0, len(landmark_errors)))
    avg_error = sum(landmark_errors) / len(landmark_errors)
    plt.figure(figsize=(12, 7))
    plt.plot(samples, landmark_errors, marker='o', linestyle='-', color='#2C3E50
    plt.axhline(y=avg_error, color='#E74C3C', linestyle='--', label=f"Average Er
    plt.title("Mean Landmark Error for the entire Database Housing {} images".fo
    plt.xlabel("Iteration Number", fontsize=14)
    plt.ylabel("Landmark Error", fontsize=14)
    plt.xticks(samples, [f"Case {i}" for i in samples], rotation=45)
    plt.grid(True, which='both', linestyle='--', linewidth=0.5)
    plt.legend(fontsize=12)
    plt.tight_layout()
    plt.savefig(os.path.join(rpth,'Landmark_Error_Plot.png'))
    plt.show();

def image_point_correspondences(images,img_size,landmarks1,landmarks2,rpth,num,s
    """
    Displays and compares point correspondences between two images using given l

    This function visualizes two images side-by-side with their respective landm
    landmarks across the images is marked with the same color for easy identific
    is designed to handle visualization for studies involving image registration
    matching is crucial.

    Parameters:
    - images (list of str): File paths to the two images (source and target imag
    - img_size (int): The size to which images should be resized, specified as w
    - landmarks1 (list of tuples): Landmark points on the first image (source im
    - landmarks2 (list of tuples): Corresponding landmark points on the second i
    - rpth (str): Path where the resultant visualization should be saved.
    - num (int): An identifier number used to differentiate the output file name
    - snum (str): Stage number or identifier to categorize the process stage.
    - disp_size (int, optional): The display size to which the image will be res
    - disp_clip (float, optional): Enabling the image to be enhanced using CLAHE

    Returns:
    - None: This function directly displays the image using matplotlib and saves
```

```
    Notes:
        The function uses OpenCV for reading and resizing images. It employs a C
        Matplotlib is used for visualizing the images and landmarks. The color m
        of landmarks; if there are more than 15 landmarks, a cyclic colormap is
        This function is particularly useful for visualizing transformations and
        similar fields where point correspondence is critical.
    """
    image1 = CLAHE_plot_cond(cv2.cvtColor(cv2.resize(cv2.imread(images[0]),(disp
    image2 = CLAHE_plot_cond(cv2.cvtColor(cv2.resize(cv2.imread(images[1]),(disp
    landmarks1 = coordinates_rescaling(landmarks1,img_size,img_size,disp_size)
    landmarks2 = coordinates_rescaling(landmarks2,img_size,img_size,disp_size)
    assert len(landmarks1) == len(landmarks2), f"points lengths are incompatible
    num_points = len(landmarks1)
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
    fig.suptitle("Stage-{} Point Correspondences".format(snum), fontsize=14, fon
    ax1.set_title('Fixed Image')
    ax2.set_title('Moving Image')
    ax1.axis('off')
    ax2.axis('off')
    ax1.imshow(image1)
    ax2.imshow(image2)
    if num_points > 15:
        cmap = plt.get_cmap('tab20')
    else:
        cmap = ListedColormap(["red", "yellow", "blue", "lime", "magenta", "indi
                               "maroon", "black", "white", "chocolate", "gray",
    colors = np.array([cmap(x) for x in range(len(landmarks1))])
    radius1, radius2 = 4, 1
    for point1, point2, color in zip(landmarks1, landmarks2, colors):
        x1, y1 = point1
        circ1_1 = plt.Circle((x1, y1), radius1, facecolor=color, edgecolor='whit
        circ1_2 = plt.Circle((x1, y1), radius2, facecolor=color, edgecolor='whit
        ax1.add_patch(circ1_1)
        ax1.add_patch(circ1_2)
        x2, y2 = point2
        circ2_1 = plt.Circle((x2, y2), radius1, facecolor=color, edgecolor='whit
        circ2_2 = plt.Circle((x2, y2), radius2, facecolor=color, edgecolor='whit
        ax2.add_patch(circ2_1)
        ax2.add_patch(circ2_2)
    plt.figtext(0.5, 0.115, "Note: {0} point correspondences were identified by
    plt.savefig(os.path.join(rpth,'Stage'+str(snum)+'Point_Correspondences'+str(
    plt.show();

def original_image_point_correspondences(images,orig_moving_image_pth,img_size,
    """
    Visualizes and saves point correspondences across three images (fixed, movin
    to aid in assessing the effectiveness of image registration processes. The f
    contrast using CLAHE for enhanced visualization and overlays landmark points

    Parameters:
    - images (list of np.array): List containing three images representing fixed
    - orig_moving_image_pth (str): Path to the original moving image, used to up
    - img_size (tuple): Original dimensions (width, height) of the images prior
    - landmarks1 (list of tuples): Coordinates of landmarks in the fixed image.
    - landmarks2 (list of tuples): Coordinates of landmarks in the original movi
    - landmarks3 (list of tuples): Coordinates of landmarks in the transformed i
    - rpth (str): Directory path where the result images will be saved.
    - num (int): Identifier to differentiate output filenames.
    - disp_size (int, optional): Target size (one dimension) for scaling images
    - disp_clip (float, optional): Enabling the image to be enhanced using CLAHE
```

```
    Raises:
    - AssertionError: If the number of landmarks in any list does not match the

    Notes:
        The images are resized to `disp_size` for display.
        Landmarks are also rescaled to match the display size.
        A colormap is applied to distinguish between different landmarks; a larg
    """
    assert len(landmarks1) == len(landmarks2) == len(landmarks3), "All landmarks
    images[1]=cv2.imread(orig_moving_image_pth) # replacing the deformed image w
    num_points = len(landmarks1)
    fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(18, 6))
    fig.suptitle("Final Registration Results by Composing Transformations Estima

    ax1.set_title('Fixed Image')
    ax2.set_title('Moving Image')
    ax3.set_title('Deformed Image')

    ax1.axis('off')
    ax2.axis('off')
    ax3.axis('off')

    ax1.imshow(CLAHE_plot_cond(cv2.cvtColor(cv2.resize(images[0],(disp_size,disp
    ax2.imshow(CLAHE_plot_cond(cv2.cvtColor(cv2.resize(images[1],(disp_size,disp
    ax3.imshow(CLAHE_plot_cond(cv2.cvtColor(cv2.resize(images[2].astype(np.uint8

    landmarks1 = coordinates_rescaling(landmarks1,img_size,img_size,disp_size)
    landmarks2 = coordinates_rescaling(landmarks2,img_size,img_size,disp_size)
    landmarks3 = coordinates_rescaling(landmarks3,img_size,img_size,disp_size)

    if num_points > 15:
        cmap = plt.get_cmap('tab20')
    else:
        cmap = ListedColormap(["red", "yellow", "blue", "lime", "magenta", "indi
                               "maroon", "black", "white", "chocolate", "gray",

    colors = np.array([cmap(x) for x in range(num_points)])
    radius1, radius2 = 4, 1

    for point1, point2, point3, color in zip(landmarks1, landmarks2, landmarks3,
        # Landmarks for Image 1
        x1, y1 = point1
        circ1_1 = plt.Circle((x1, y1), radius1, facecolor=color, edgecolor='whit
        circ1_2 = plt.Circle((x1, y1), radius2, facecolor=color, edgecolor='whit
        ax1.add_patch(circ1_1)
        ax1.add_patch(circ1_2)

        # Landmarks for Image 2
        x2, y2 = point2
        circ2_1 = plt.Circle((x2, y2), radius1, facecolor=color, edgecolor='whit
        circ2_2 = plt.Circle((x2, y2), radius2, facecolor=color, edgecolor='whit
        ax2.add_patch(circ2_1)
        ax2.add_patch(circ2_2)

        # Landmarks for Image 3
        x3, y3 = point3
        circ3_1 = plt.Circle((x3, y3), radius1, facecolor=color, edgecolor='whit
        circ3_2 = plt.Circle((x3, y3), radius2, facecolor=color, edgecolor='whit
        ax3.add_patch(circ3_1)
```

```
        ax3.add_patch(circ3_2)

    plt.savefig(os.path.join(rpth, 'Final_Registration_Results_for_case' + str(n
    plt.show();
```

In [12]:

```
def transform_points_affine(moving_points, affine_matrix):
    """
    Transform the moving points using the given affine matrix.

    Parameters:
    - moving_points: List of (x, y) tuples
    - affine_matrix: (3x3) affine matrix

    Returns:
    - List of (x, y) tuples representing transformed points
    """
    points_array = np.array(moving_points)
    homogeneous_points = np.hstack([points_array, np.ones((len(moving_points), 1
    transformed_points = np.dot(homogeneous_points, affine_matrix.T)
    return [tuple(point) for point in transformed_points[:, :2]]

def transform_points_homography(moving_points, homography_matrix):
    """
    Transform the moving points using the given homography matrix.

    Parameters:
    - moving_points: List of (x, y) tuples
    - homography_matrix: (3x3) homography matrix

    Returns:
    - List of (x, y) tuples representing transformed points
    """
    points_array = np.array(moving_points)
    homogeneous_points = np.hstack([points_array, np.ones((len(moving_points), 1
    transformed_points = np.dot(homogeneous_points, homography_matrix.T)
    transformed_points /= transformed_points[:, 2][:, np.newaxis]  # Normalize b
    return [tuple(point[:2]) for point in transformed_points]

def transform_points_third_order_polynomial(moving_points, coefficients):
    """
    Transform the moving points using the given third-order polynomial coefficie

    Parameters:
    - moving_points: List of (x, y) tuples
    - coefficients: Array of 20 coefficients for the third-order polynomial tran

    Returns:
    - List of (x, y) tuples representing transformed points
    """
    if len(coefficients) != 20:
        raise ValueError("Coefficients should have a shape of (20,).")

    # Extract the coefficients
    a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, \
    a11, a12, a13, a14, a15, a16, a17, a18, a19, a20 = coefficients

    transformed_points = []
    for x, y in moving_points:
        # Compute new x' and y' for each point using third-order polynomial
        x_prime = (a1*x**3 + a2*x**2*y + a3*x*y**2 + a4*y**3 +
```

```
                    a5*x**2 + a6*x*y + a7*y**2 + a8*x + a9*y + a10)
        y_prime = (a11*x**3 + a12*x**2*y + a13*x*y**2 + a14*y**3 +
                    a15*x**2 + a16*x*y + a17*y**2 + a18*x + a19*y + a20)
        transformed_points.append((x_prime, y_prime))

    return transformed_points

def transform_points_quadratic(points, coefficients):
    """
    Applies a quadratic transformation to a set of 2D points based on the provid
    typically used in image processing and computer vision tasks to deform point

    Parameters:
    - points (list of tuples): A list of points, where each point is represented
    - coefficients (list): A list of 12 coefficients for the quadratic transform

    Returns:
    - list: A list of tuples representing the deformed points.

    Raises:
    - ValueError: If the number of coefficients is not equal to 12, as the quadr

    Notes:
        This function uses a quadratic transformation defined as:
        x' = a1*x + a2*y + a3*x*y + a4*x^2 + a5*y^2 + a6
        y' = a7*x + a8*y + a9*x*y + a10*x^2 + a11*y^2 + a12
        where `x, y` are the original coordinates and `x', y'` are the transform
        The coefficients must be specified in the order [a1, a2, a3, a4, a5, a6,
    """
    if len(coefficients) != 12:
        raise ValueError("Coefficients should have a shape of (12,).")

    a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12 = coefficients

    deformed = []
    for x, y in points:
        x_prime = a1*x + a2*y + a3*x*y + a4*x**2 + a5*y**2 + a6
        y_prime = a7*x + a8*y + a9*x*y + a10*x**2 + a11*y**2 + a12
        deformed.append((x_prime, y_prime))

    return deformed

def compute_landmark_error_fixed_space(polynomial_matrix,fixed_points,moving_poi
    """
    Compute the landmark error between fixed points and transformed moving point

    Parameters:
    - fixed_points: List of (x, y) tuples in the fixed image.
    - moving_points: List of (x, y) tuples in the moving image.
    - polynomial_matrix: (3x3) matrix used to transform points using a third-ord
    - image_size: The original size of the images.
    - new_image_size: The size of the images after rescaling.

    Returns:
    - mle: Mean Landmark Error.
    """
    transformed_points = transform_points_third_order_polynomial(moving_points,
    transformed_points = coordinates_rescaling_high_scale(transformed_points,new
    errors = np.linalg.norm(np.array(fixed_points) - transformed_points, axis=1)
    mle = np.mean(errors)
```

```
    return mle

def compute_landmark_error(fixed_points,fixed_image_size,moving_points,moving_im
    """
    Calculates the mean landmark error between fixed points and transformed movi
    after rescaling to a new image size. This function is primarily used in imag
    to measure the accuracy of image registration by quantifying the displacemen

    Parameters:
    - fixed_points (list of tuples): Coordinates of landmark points in the fixed
    - fixed_image_size (tuple): The original size (width, height) of the fixed i
    - moving_points (list of tuples): Coordinates of landmark points in the movi
    - moving_image_size (tuple): The original size (width, height) of the moving
    - new_image_size (int): The size to which both sets of points will be resize

    Returns:
    - float: The mean landmark error calculated as the average Euclidean distanc
          landmarks after rescaling to the new image size.

    Notes:
        The function first rescales the coordinates of both fixed and moving poi
        It then calculates the Euclidean distance between the corresponding resc
        This metric is useful for evaluating the precision of image registration
    """
    rescaled_fixed_points = coordinates_rescaling_high_scale(fixed_points,new_im
    rescaled_moving_points = coordinates_rescaling_high_scale(moving_points,new_
    errors = np.linalg.norm(np.array(rescaled_fixed_points) - rescaled_moving_po
    mle = np.mean(errors)
    return mle

def compute_third_order_polynomial_matrix(landmarks1, landmarks2):
    """
    Compute coefficients for the third-order polynomial transformation.

    Parameters:
    - landmarks1 (list): List of (x, y) tuples of landmarks in the first image.
    - landmarks2 (list): List of (x, y) tuples of landmarks in the second image.

    Returns:
    - np.array: Coefficients of the third-order polynomial transformation.
    """
    if len(landmarks1) != len(landmarks2) or len(landmarks1) < 10:
        raise ValueError("Both landmarks should have the same number of points,

    A = []
    B = []

    for (x, y), (x_prime, y_prime) in zip(landmarks1, landmarks2):
        # For x'
        A.append([x**3, x**2 * y, x * y**2, y**3, x**2, x * y, y**2, x, y, 1, 0,
        # For y'
        A.append([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, x**3, x**2 * y, x * y**2, y**3,

        B.extend([x_prime, y_prime])

    A = np.array(A)
    B = np.array(B)

    # Solve the linear system
    coefficients, _, _, _ = np.linalg.lstsq(A, B, rcond=None)
```

```
    return coefficients  # The shape of coefficients is (20,)

def compute_quadratic_matrix(landmarks1, landmarks2):
    """
    Compute the quadratic matrix using provided landmarks.

    Parameters:
    - landmarks1: List of (x, y) tuples from the source image.
    - landmarks2: List of (x, y) tuples from the target image.

    Returns:
    - 3x3 homography matrix.
    """
    if len(landmarks1) != len(landmarks2) or len(landmarks1) < 6:
        raise ValueError("Both landmarks should have the same number of points,

    A = []
    B = []

    for (x, y), (x_prime, y_prime) in zip(landmarks1, landmarks2):
        A.append([x, y, x*y, x*x, y*y, 1, 0, 0, 0, 0, 0, 0])
        A.append([0, 0, 0, 0, 0, 0, x, y, x*y, x*x, y*y, 1])

        B.append(x_prime)
        B.append(y_prime)

    A = np.array(A)
    B = np.array(B)

    # Solve the linear system
    coefficients, _, _, _ = np.linalg.lstsq(A, B, rcond=None)

    return coefficients

def compute_homography_matrix(landmarks1, landmarks2):
    """
    Compute the homography matrix using provided landmarks.

    Parameters:
    - landmarks1: List of (x, y) tuples from the source image.
    - landmarks2: List of (x, y) tuples from the target image.

    Returns:
    - 3x3 homography matrix.
    """
    homography_matrix, _ = cv2.findHomography(np.array(landmarks1), np.array(lan
    return homography_matrix

def transform_points_third_order_polynomial_matrix(landmarks1, landmarks2,img_si
    """
    Computes a third-order polynomial transformation matrix based on rescaled la
    to another. This transformation is typically used for tasks like geometric t
    alignment or registration of image features is necessary.

    Parameters:
    - landmarks1 (list of tuples): List of original landmark points in the sourc
    - landmarks2 (list of tuples): List of corresponding landmark points in the
                                   The points in landmarks2 should correspond one-
    - img_size (int): Original size of the images from which the landmarks were
```

```
                    rescale points for accurate computation of the transformatio
    - new_img_size (int): New size to which the points will be rescaled before c
                        This should reflect the size of the image space into whi

    Returns:
    - numpy.ndarray: A transformation matrix which can be used to map the points
                      to the space defined by landmarks2. The matrix is represe
                      corresponding to the terms of a third-order polynomial.

    Notes:
        Ensure that the number of points in landmarks1 and landmarks2 are equal
        This function involves rescaling coordinates, calculating a transformati
        tasks where geometric transformations are necessary for alignment and re
    """
    landmarks1 = coordinates_rescaling(landmarks1,img_size,img_size,new_img_size
    landmarks2 = coordinates_rescaling(landmarks2,img_size,img_size,new_img_size
    third_order_polynomial_matrix  = compute_third_order_polynomial_matrix(landm
    return third_order_polynomial_matrix

def transform_points_quadratic_matrix(landmarks1, landmarks2,img_size,new_img_si
    """
    Computes a quadratic transformation matrix based on rescaled landmarks from

    This function rescales the input landmarks from their original dimensions (i
    It then calculates a quadratic transformation matrix that describes how poin
    can be transformed to align with the second set (landmarks2). This matrix co
    to images or coordinates.

    Parameters:
    - landmarks1 (list of tuples): List of (x, y) tuples representing original l
    - landmarks2 (list of tuples): List of (x, y) tuples representing target lan
                                   corresponding to landmarks1.
    - img_size (int): The original size (width and height, assumed square) of th
    - new_img_size (int): The new size (width and height, assumed square) to whi
                        before computing the transformation matrix.

    Returns:
    - numpy.ndarray: A matrix that contains the coefficients of the quadratic tr
                      to transform points from the source image to the target i

    Notes:
        Ensure that the number of points in landmarks1 and landmarks2 are equal
        This function is essential in image processing tasks where precise trans
    """
    landmarks1 = coordinates_rescaling(landmarks1,img_size,img_size,new_img_size
    landmarks2 = coordinates_rescaling(landmarks2,img_size,img_size,new_img_size
    quadratic_matrix = compute_quadratic_matrix(landmarks1, landmarks2)
    return quadratic_matrix


def warp_image_third_order_polynomial(image, coefficients):
    """
    Applies a third-order polynomial transformation to an image using provided c

    Parameters:
    - image (numpy.ndarray): The image to deform, provided as a numpy array. The
                           two-dimensional (grayscale image) or three-dimensiona
    - coefficients (list or array): An array of 20 coefficients for the third-or

    Raises:
```

```
    - ValueError: If the number of coefficients provided is not 20, an error is
                of exactly 20 coefficients to perform the transformation.

    Returns:
    - numpy.ndarray: The deformed image as a numpy array of the same shape as th

    Notes:
        The deformation is defined by a polynomial transformation that adjusts t
        based on the polynomial defined by the coefficients.
        This function supports both grayscale and color images. For color images
        to each color channel independently.
        The transformation involves calculating new pixel positions and mapping
        to these new positions using spline interpolation of order 1.
    """
    if len(coefficients) != 20:
        raise ValueError("Coefficients should have a shape of (20,).")

    # Extract the coefficients
    a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, \
    a11, a12, a13, a14, a15, a16, a17, a18, a19, a20 = coefficients

    # Check if the image is grayscale or colored
    if len(image.shape) == 2:
        height, width = image.shape
        output = np.zeros((height, width))
        channels = 1
        image = image[:, :, np.newaxis]  # add an additional dimension for consi
    else:
        height, width, channels = image.shape
        output = np.zeros((height, width, channels))

    # Generate the coordinates
    coordinates = np.indices((height, width))
    x_coords = coordinates[1]
    y_coords = coordinates[0]

    # Compute new x' and y' for every x and y using third-order polynomial
    x_prime = (a1*x_coords**3 + a2*x_coords**2*y_coords + a3*x_coords*y_coords**
               a5*x_coords**2 + a6*x_coords*y_coords + a7*y_coords**2 + a8*x_coc
    y_prime = (a11*x_coords**3 + a12*x_coords**2*y_coords + a13*x_coords*y_coord
               a15*x_coords**2 + a16*x_coords*y_coords + a17*y_coords**2 + a18*x

    # Map the old image pixels to the new deformed positions
    for c in range(channels):  # for each channel
        output[:, :, c] = map_coordinates(image[:, :, c], [y_prime, x_prime], or

    if channels == 1:
        return output[:, :, 0]  # return as 2D grayscale image
    else:
        return output

def warp_image_quadratic_matrix(image, coefficients):
    """
    Applies a quadratic transformation to deform an image using provided coeffic

    Parameters:
    - image (numpy.ndarray): The image to deform, represented as a numpy array.
                             either two-dimensional (grayscale image) or three-dim
    - coefficients (list or array): A list or array of 12 coefficients defining
```

```
    Raises:
    - ValueError: If the number of coefficients provided is not equal to 12, rai
                that exactly 12 coefficients are required for the transformation

    Returns:
    - numpy.ndarray: The deformed image as a numpy array of the same shape as th

    Notes:
        The deformation involves calculating new pixel coordinates using the qua
        by the coefficients and then mapping the original pixel values to these
        The function checks if the image is in grayscale or color and processes
        The mapping of pixels uses spline interpolation of order 1 for accuracy
        the transformed coordinates with zeros.
    """
    if len(coefficients) != 12:
        raise ValueError("Coefficients should have a shape of (12,).")

    a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12 = coefficients

    # Check if the image is grayscale or colored
    if len(image.shape) == 2:
        height, width = image.shape
        output = np.zeros((height, width))
        channels = 1
        image = image[:, :, np.newaxis]  # add an additional dimension for consi
    else:
        height, width, channels = image.shape
        output = np.zeros((height, width, channels))

    # Generate the coordinates
    coordinates = np.indices((height, width))
    x_coords = coordinates[1]
    y_coords = coordinates[0]

    # Compute new x' and y' for every x and y
    x_prime = a1*x_coords + a2*y_coords + a3*x_coords*y_coords + a4*x_coords**2
    y_prime = a7*x_coords + a8*y_coords + a9*x_coords*y_coords + a10*x_coords**2

    # Map the old image pixels to the new deformed positions
    for c in range(channels):  # for each channel
        output[:, :, c] = map_coordinates(image[:, :, c], [y_prime, x_prime], or

    if channels == 1:
        return output[:, :, 0]  # return as 2D grayscale image
    else:
        return output


def compute_third_order_polynomial_matrix_and_plot(images, img_size, landmarks1,
    """
    Computes a third-order polynomial transformation matrix based on landmark co
    between two images and applies this transformation to align one image with a
    also displays and saves the original and transformed images, enhancing their
    visibility.

    Parameters:
    - images (list of str): Paths to the source and target images.
    - img_size (int): The dimensions (height and width) to which the images shou
    - landmarks1 (list of tuples): Coordinates of landmarks in the source image.
    - landmarks2 (list of tuples): Corresponding coordinates of landmarks in the
```

```
    - rpth (str): Path to the directory where the resultant images will be saved
    - num (int): An identifier number for differentiating the output file names.
    - snum (int): Stage number for referencing in output.
    - disp_clip (float, optional): Clipping limit for the CLAHE algorithm, used

    Raises:
    - ValueError: If the list of landmarks from the source image is empty.

    Returns:
    tuple: Contains three elements:
        - imags (list of np.array): The original fixed and moving images along w
        - imgs (list of str): Paths to the saved output images.
        - coefficients (np.array): Coefficients of the third-order polynomial us

    Notes:
        This function is suited for complex registration tasks where finer contr
        The transformation matrix is applied to the target image to align it wit
        The images are displayed and saved with enhanced contrast to aid in visu
    """
    imags,imgs = [],[]
    img1 = cv2.resize(cv2.imread(images[0]), (img_size, img_size))
    img2 = cv2.resize(cv2.imread(images[1]), (img_size, img_size))

    imags.append(img1)
    imags.append(img2)

     # Check if the list is not empty
    if not landmarks1:  raise ValueError("Input list cannot be empty")

    # Check and delte the temporary folder if it exists
    if os.path.exists(os.path.join(os.getcwd(),'temp_dir')): shutil.rmtree(os.pa

    # Compute the third-order polynomial transformation matrix
    coefficients = compute_third_order_polynomial_matrix(landmarks1, landmarks2)
    coefficients_for_transform = compute_third_order_polynomial_matrix(landmarks

    # Apply the transformation using third-order polynomial
    transformed_image = warp_image_third_order_polynomial(img2, coefficients_for
    imags.append(transformed_image)

    # Display and save the images
    fig, axs = plt.subplots(1, 3, figsize=(18, 6))
    fig.suptitle("Stage-{} Results: Registration Using Third Order Polynomial Tr

    axs[0].imshow(CLAHE_plot_cond(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB),disp_cli
    axs[0].set_title('Fixed Image')
    axs[0].axis('off')

    axs[1].imshow(CLAHE_plot_cond(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB),disp_cli
    axs[1].set_title('Moving Image')
    axs[1].axis('off')

    axs[2].imshow(CLAHE_plot_cond(cv2.cvtColor(transformed_image.astype(np.uint8
    axs[2].set_title('Deformed Image')
    axs[2].axis('off')

    plt.show();

    # saving intermediary results for better visualization
    imgs.append(os.path.join(rpth, 'Deformed_Image_' + str(num) + '_.png'))
```

```
    imgs.append(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'))
    cv2.imwrite(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'), img1)
    cv2.imwrite(os.path.join(rpth, 'Moving_' + str(num) + '_.png'), img2)
    cv2.imwrite(os.path.join(rpth,'Deformed_Image_'+str(num)+'_.png'),transforme
    return imags,imgs,coefficients

def compute_affine_matrix_and_plot(images,img_size,landmarks1, landmarks2,rpth,n
    """
    Computes an affine transformation matrix based on provided landmarks from tw
    this transformation to visually compare the source, target, and transformed

    This function computes the affine transformation matrix that best maps the s
    with the target image using landmark correspondences. It then applies this t
    the source image and displays the original (source and target) and transform
    The images are enhanced using CLAHE for better visibility and are saved to t

    Parameters:
    - images (list of str): File paths for the source and target images.
    - img_size (int): The size (width and height) to which the images will be re
    - landmarks1 (list of tuples): Landmark points (x, y) on the source image.
    - landmarks2 (list of tuples): Corresponding landmark points (x, y) on the t
    - rpth (str): Directory path where the resultant images will be saved.
    - num (int): Identifier number used to differentiate the output file names.
    - snum (int): Stage number for referencing in output.
    - disp_clip (float, optional): Clipping limit for the CLAHE algorithm, used

    Returns:
    - tuple: Contains two items:
        - imgs (list of str): Paths to the saved images.
        - affine_matrix (numpy.ndarray): The computed affine transformation matr

    Raises:
    - ValueError: If the landmarks list is empty, indicating insufficient data t

    Notes:
        The affine transformation matrix is computed using a least squares metho
        This function is useful for tasks in image registration where visual com
        Enhanced contrast is used to aid in the visual assessment of image regis
    """
    imgs,imags=[],[]
    img1 = cv2.resize(cv2.imread(images[0]),(img_size,img_size))
    img2 = cv2.resize(cv2.imread(images[1]),(img_size,img_size))

    imags.append(img1)
    imags.append(img2)

    # Check if the list is not empty
    if not landmarks1:  raise ValueError("Input list cannot be empty")

    # Check and delte the temporary folder if it exists
    if os.path.exists(os.path.join(os.getcwd(),'temp_dir')): shutil.rmtree(os.pa

    # Create the array with the specified format
    A = np.array([[xs, ys, 1] for xs, ys in landmarks1])

    X = np.array([xt for xt, yt in landmarks2])
    Y = np.array([yt for xt, yt in landmarks2])

    # Solve for the variables x1, y1, and z1
    sol1 = np.dot(np.dot(np.linalg.inv(np.dot(A.T,A)),A.T),X)
```

```
    sol2 = np.dot(np.dot(np.linalg.inv(np.dot(A.T,A)),A.T),Y)
    # Extract the variables
    x1, y1, z1 = sol1
    x2, y2, z2 = sol2
    affine_matrix = np.array([[x1,y1,z1],
                              [x2,y2,z2],
                              [0, 0, 1]])
    print("Affine Matrix:")
    print(affine_matrix)

    # Ensure the affine matrix is of type float32
    affine_matrix = affine_matrix.astype(np.float32)

    # Use only the top two rows for cv2.warpAffine
    affine_for_warp = affine_matrix[:2]

    # Apply the affine transformation using cv2.warpAffine
    transformed_image = cv2.warpAffine(img2, affine_for_warp, (img2.shape[1], im
    imags.append(transformed_image)

    # Display and save the images
    fig, axs = plt.subplots(1, 3, figsize=(18, 6))
    fig.suptitle("Stage-{} Results: Registration Using Affine Transformation".fo

    axs[0].imshow(CLAHE_plot_cond(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB),disp_cli
    axs[0].set_title('Fixed Image')
    axs[0].axis('off')

    axs[1].imshow(CLAHE_plot_cond(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB),disp_cli
    axs[1].set_title('Moving Image')
    axs[1].axis('off')

    axs[2].imshow(CLAHE_plot_cond(cv2.cvtColor(transformed_image.astype(np.uint8
    axs[2].set_title('Deformed Image')
    axs[2].axis('off')

    plt.show();

    # saving intermediary results for better visualization
    imgs.append(os.path.join(rpth, 'Deformed_Image_' + str(num) + '_.png'))
    imgs.append(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'))
    cv2.imwrite(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'), img1)
    cv2.imwrite(os.path.join(rpth, 'Moving_' + str(num) + '_.png'), img2)
    cv2.imwrite(os.path.join(rpth,'Deformed_Image_'+str(num)+'_.png'),transforme
    return imgs,imags,affine_matrix

def compute_quadratic_matrix_and_plot(images,img_size,landmarks1, landmarks2,rpt
    """
    Computes a quadratic transformation matrix from source to target landmarks a
    to the source image. The transformed source image is displayed alongside the
    and all images are saved to disk.

    This function takes pairs of corresponding landmarks from the source and tar
    transformation matrix. This matrix is then used to warp the source image to
    result, along with the original images, is displayed and saved for compariso

    Parameters:
    - images (list of str): File paths for the source and target images.
    - img_size (int): The size (width and height) to which the images will be re
    - landmarks1 (list of tuples): Landmark points (x, y) on the source image.
```

```
    - landmarks2 (list of tuples): Corresponding landmark points (x, y) on the t
    - rpth (str): Directory path where the resultant images will be saved.
    - num (int): Identifier number used to differentiate the output file names.
    - cll (float, optional): Clipping limit for the CLAHE algorithm used in cont
    - snum (int): Stage number used for displaying in the title of the plot.
    - disp_clip (float, optional): Clipping limit for the CLAHE algorithm, used

    Returns:
    - tuple: Contains three items:
        - imgs (list of str): File paths where the output images are saved.
        - imags (list of np.array): List containing the numpy arrays of the orig
        - quadratic_matrix (numpy.ndarray): The computed quadratic transformatio

    Raises:
    - AssertionError: If the number of points in `landmarks1` and `landmarks2` a
                    number of points is required for matrix computation.

    Notes:
        The function uses OpenCV for image processing tasks such as reading, res
        The quadratic transformation matrix is computed using a least squares me
        Matplotlib is used for visualizing the before and after effects of the t
        This function is particularly useful in applications such as image regis
    """
    imgs,imags=[],[]
    img1 = cv2.resize(cv2.imread(images[0]),(img_size,img_size))
    img2 = cv2.resize(cv2.imread(images[1]),(img_size,img_size))

    imags.append(img1)
    imags.append(img2)

    # Check if the list is not empty
    if not landmarks1:  raise ValueError("Input list cannot be empty")

    # Check and delte the temporary folder if it exists
    if os.path.exists(os.path.join(os.getcwd(),'temp_dir')): shutil.rmtree(os.pa

    # Ensure the quadratic matrix is of type float32
    quadratic_matrix = compute_quadratic_matrix(landmarks1, landmarks2)

    quadratic_matrix_for_image_deformed = compute_quadratic_matrix(landmarks2, l

    print("quadratic Matrix:")
    print(quadratic_matrix)

    # Apply the quadratic transformation using cv2.warpquadratic
    transformed_image =  warp_image_quadratic_matrix(img2, quadratic_matrix_for_
    transformed_image = cv2.resize(transformed_image,  (img2.shape[1], img2.shap
    imags.append(transformed_image)

    # Display and save the images
    fig, axs = plt.subplots(1, 3, figsize=(18, 6))
    fig.suptitle("Stage-{} Results: Registration Using Quadratic Transformation"

    axs[0].imshow(CLAHE_plot_cond(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB),disp_cli
    axs[0].set_title('Fixed Image')
    axs[0].axis('off')

    axs[1].imshow(CLAHE_plot_cond(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB),disp_cli
    axs[1].set_title('Moving Image')
    axs[1].axis('off')
```

```
    axs[2].imshow(CLAHE_plot_cond(cv2.cvtColor(transformed_image.astype(np.uint8
    axs[2].set_title('Deformed Image')
    axs[2].axis('off')

    plt.show();

    # saving intermediary results for better visualization
    imgs.append(os.path.join(rpth, 'Deformed_Image_' + str(num) + '_.png'))
    imgs.append(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'))
    cv2.imwrite(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'), img1)
    cv2.imwrite(os.path.join(rpth, 'Moving_' + str(num) + '_.png'), img2)
    cv2.imwrite(os.path.join(rpth,'Deformed_Image_'+str(num)+'_.png'),transforme
    return imgs,imags,quadratic_matrix

def compute_homography_matrix_and_plot(images, img_size, landmarks1, landmarks2,
    """
    Computes the homography transformation matrix based on landmark correspondenc
    and applies this transformation to the source image. The function displays t
    target images along with the transformed source image. It also saves these i

    Parameters:
    - images (list of str): Paths to the source and target images.
    - img_size (int): The size to which both images will be resized.
    - landmarks1 (list of tuples): Landmark points (x, y) from the source image.
    - landmarks2 (list of tuples): Corresponding landmark points (x, y) from the
    - rpth (str): The directory path where the resultant images will be saved.
    - num (int): An identifier number used to differentiate the output file name
    - snum (int): Stage number used for displaying in the title of the plot.
    - disp_clip (float, optional): Clipping limit for the CLAHE algorithm, used

    Returns:
    - tuple: A tuple containing the paths to the saved images, a list of image a
           and the computed homography matrix.

    Raises:
    - ValueError: If the list of landmarks is empty, indicating that there are n

    Notes:
        The function uses OpenCV for image reading, resizing, and applying the h
        Matplotlib is used for displaying the images.
        Ensure the landmarks are accurately defined as their correspondence dire
        Homography transformations are particularly useful for applications in i
    """
    imgs,imags=[],[]
    img1 = cv2.resize(cv2.imread(images[0]),(img_size,img_size))
    img2 = cv2.resize(cv2.imread(images[1]),(img_size,img_size))

    imags.append(img1)
    imags.append(img2)

    # Check if the list is not empty
    if not landmarks1:  raise ValueError("Input list cannot be empty")

    # Check and delte the temporary folder if it exists
    if os.path.exists(os.path.join(os.getcwd(),'temp_dir')): shutil.rmtree(os.pa

    # Compute homography matrix
    homography_matrix = compute_homography_matrix(landmarks1, landmarks2)
```

```
    print("Homography Matrix:")
    print(homography_matrix)

    # Ensure the affine matrix is of type float32
    homography_matrix = homography_matrix.astype(np.float32)

    # Apply the homography transformation using cv2.warpPerspective
    transformed_image=cv2.warpPerspective(img2, homography_matrix, (img2.shape[1
    imags.append(transformed_image)

    # Display and save the images
    fig, axs = plt.subplots(1, 3, figsize=(18, 6))
    fig.suptitle("Stage-{} Results: Registration Using Homography Transformation

    axs[0].imshow(CLAHE_plot_cond(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB),disp_cli
    axs[0].set_title('Fixed Image')
    axs[0].axis('off')

    axs[1].imshow(CLAHE_plot_cond(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB),disp_cli
    axs[1].set_title('Moving Image')
    axs[1].axis('off')

    axs[2].imshow(CLAHE_plot_cond(cv2.cvtColor(transformed_image.astype(np.uint8
    axs[2].set_title('Deformed Image')
    axs[2].axis('off')

    plt.show();

    # saving intermediary results for better visualization
    imgs.append(os.path.join(rpth, 'Deformed_Image_' + str(num) + '_.png'))
    imgs.append(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'))
    cv2.imwrite(os.path.join(rpth, 'Fixed_' + str(num) + '_.png'), img1)
    cv2.imwrite(os.path.join(rpth, 'Moving_' + str(num) + '_.png'), img2)
    cv2.imwrite(os.path.join(rpth,'Deformed_Image_'+str(num)+'_.png'),transforme
    return imgs,imags,homography_matrix
```

In [13]:

```
def landmark_error(point, transformed_point):
    """
    Computes the Euclidean distance between the original point and the transform

    Parameters:
    - point (tuple): Original point (x, y).
    - transformed_point (tuple): Transformed point (x, y).

    Returns:
    - float: Euclidean distance.
    """
    return np.linalg.norm(np.array(point) - np.array(transformed_point))

def estimate_affine_transformation(points):
    """
    Estimates the affine transformation matrix using point correspondences.

    Parameters:
    - points (np.array): Array of point correspondences.

    Returns:
    - np.array: Affine transformation matrix.
    """
    src_pts = np.float32([point[0] for point in points])
```

```
    dst_pts = np.float32([point[1] for point in points])
    affine_matrix, _ = cv2.estimateAffinePartial2D(src_pts, dst_pts)
    return affine_matrix

def estimate_homography_matrix(points):
    """
    Estimates the homography matrix given a set of point correspondences.

    Parameters:
    - points: A list of tuples, where each tuple contains two (x, y) tuples.
              The first tuple in each pair is from the first set of points (set1
              and the second tuple is the corresponding point in the second set

    Returns:
    - homography_matrix: The estimated (3x3) homography matrix.
    """

    # Separate the points into two sets
    set1 = [point[0] for point in points]
    set2 = [point[1] for point in points]

    # Convert to numpy arrays
    set1 = np.array(set1, dtype=np.float32)
    set2 = np.array(set2, dtype=np.float32)

    # Estimate the homography matrix
    homography_matrix, _ = cv2.findHomography(set1, set2, cv2.RANSAC)

    return homography_matrix

def remove_outliers_based_on_error_affine(set1, set2, threshold=20):
    """
    Filters out outlier point pairs from two sets of points by applying an affin
    removing pairs that have an error greater than a specified threshold. The fu
    an affine transformation matrix based on all given point pairs. Each point i
    transformed using this matrix, and the error is calculated as the Euclidean
    transformed point and the corresponding point in the second set. Points with
    threshold are considered outliers and are excluded from the results.

    Parameters:
    - set1 (list of tuples): A list of (x, y) tuples representing coordinates of
    - set2 (list of tuples): A list of (x, y) tuples representing corresponding
                             second image. The indices in `set1` and `set2` must c
    - threshold (float, optional): The maximum allowed error distance between th
                                   points for them to be considered inliers. Defau

    Returns:
    - tuple of lists: Returns two lists (updated_set1, updated_set2) containing
                      `set1` and `set2` respectively.
    Notes:
        It is critical that `set1` and `set2` are of equal length and that the p
        as any misalignment could result in incorrect calculations and poor resu
        This function is typically used in image processing and computer vision
        transformation of point sets between images is required, particularly in
    """
    points = list(zip(set1, set2))
    affine_matrix = estimate_affine_transformation(points)
    updated_set1 = []
    updated_set2 = []
```

```
    for point1, point2 in zip(set1, set2):
        transformed_point = transform_points_affine([point1], affine_matrix)[0]
        error = landmark_error(point2, transformed_point)

        if error <= threshold:
            updated_set1.append(point1)
            updated_set2.append(point2)

    return updated_set1, updated_set2

def remove_outliers_based_on_error_homography(set1, set2, threshold=20):
    """
    Filters out outlier point pairs from two sets of points by applying a homogr
    and removing pairs that have an error greater than a specified threshold. Th
    a homography transformation matrix based on all given point pairs. Each poin
    transformed using this matrix, and the error is calculated as the Euclidean
    transformed point and the corresponding point in the second set. Points with
    threshold are considered outliers and are excluded from the results.

    Parameters:
    - set1 (list of tuples): A list of (x, y) tuples representing coordinates of
    - set2 (list of tuples): A list of (x, y) tuples representing corresponding
                             second image. The indices in `set1` and `set2` must c
    - threshold (float, optional): The maximum allowed error distance between th
                                   points for them to be considered inliers. Defau

    Returns:
    - tuple of lists: Returns two lists (updated_set1, updated_set2) containing
                      `set1` and `set2` respectively.
    Notes:
        Ensure that `set1` and `set2` are of equal length and that the points co
        as any misalignment could result in incorrect calculations and poor resu
        This function is typically used in image processing and computer vision
        alignment and transformation of point sets between images are required,
        like panorama stitching and object tracking.
    """
    points = list(zip(set1, set2))
    homography_matrix = estimate_homography_matrix(points)
    updated_set1 = []
    updated_set2 = []

    for point1, point2 in zip(set1, set2):
        transformed_point = transform_points_homography([point1], homography_mat
        error = landmark_error(point2, transformed_point)

        if error <= threshold:
            updated_set1.append(point1)
            updated_set2.append(point2)

    return updated_set1, updated_set2

def filter_outlier_cond(computed,original,criteria='affine', thresh=20):
    """
    Filters out outliers based on a specified condition.

    This function processes two sets of points (computed and original) and filte

    Parameters:
    - computed (list of tuples): List of computed points as (x, y) coordinates.
    - original (list of tuples): List of original points as (x, y) coordinates t
```

```
    - criteria (str, optional): The criteria to use for filtering outliers. Opti
    - thresh (int, optional): Threshold value used in the outlier removal proces

    Returns:
    - list: A list containing the filtered computed points after outlier removal
    - list: A list containing the filtered original points after outlier removal

    Raises:
    - AssertionError: If the length of the computed points is not 3.

    Notes:
        If 'homography' is chosen as the criteria, the function estimates a homo
        If 'affine' is chosen, it removes outliers based on affine transformatio
    """
    assert len(computed) >= 3
    if criteria=='homography':
        computed,original = estimate_homography_matrix(computed,original,thresh)
    else:
        computed,original = remove_outliers_based_on_error_affine(computed,origi
    return computed,original
```

In [14]:

```
def main_initialization(images,N,img_size,max_dist,offset,window_size,clip):
    """
    Initializes image processing by applying CLAHE if specified, extracting keyp
    and computing the Discrete Fourier Transform (DFT) for the given images.

    Parameters:
        - images (list of str): List of image file paths that need processing.
        - N (int): Number of keypoints to detect or random points to select.
        - img_size (tuple of int): The dimensions (width, height) to which image
        - max_dist (float): Maximum distance between keypoints for the SIFT algo
        - offset (float): Offset used in the selection of random points.
        - window_size (int): Size of the window used in random point selection.
        - clip (float): Clipping limit for the CLAHE algorithm; if greater than

    Returns:
        - tuple:
            - images (list of np.array): The list of images after processing, po
            - pts(list of tuples): the list of detected points after applying SI
            - dft (np.array): The result of the Discrete Fourier Transform appli

    Notes:
        The function begins by extracting SIFT keypoints from the first image an
        It then applies CLAHE if the clipping limit is specified and computes th
    """
    pts = SIFT_top_n_keypoints(images[0],N,img_size,max_dist)
    pts = pts+select_random_points(images[0],N,img_size,offset,window_size)
    if clip > 0:
        images = CLAHE_Images(images, clip = clip)
    dft = DFT(images,img_size,pts)
    return images,pts,dft

def CLAHE_Images(imags,clip):
    """
    Applies Contrast Limited Adaptive Histogram Equalization (CLAHE) to a list o
    their contrast. This method is particularly useful for improving the visibil
    that suffer from poor contrast.

    Parameters:
    - imags (list of str): List of paths to the image files that need contrast e
```

```
    - clip (float): Clip limit for the CLAHE algorithm, which sets the threshold
                  The higher the clip limit, the more aggressive the contrast en

    Returns:
    - list of str: Returns a list of paths to the saved CLAHE-processed images.
                saved with a "CLAHE_" prefix in its filename to distinguish it

    Notes:
        This function uses OpenCV's `createCLAHE` method to apply the CLAHE algo
        first converted to grayscale as CLAHE is typically applied to single-cha
        visualization of detail.
        The images are processed in-place and saved in the same directory as the
        prefixed to their original filenames.
        It is recommended to adjust the `clip` parameter based on the specific r
        content and the desired level of contrast enhancement.
    """
    imgs=[]
    img_dir = os.path.join(os.getcwd(),'temp_dir')
    clahe = cv2.createCLAHE(clipLimit=clip, tileGridSize=(8, 8))
    os.makedirs(os.path.join(os.getcwd(),'temp_dir') ,exist_ok=True)
    for img in imags:
      fn,_ = os.path.splitext(os.path.basename(img))
      ifn = os.path.join(img_dir,'CLAHE'+'_'+str(fn)+'.png')
      imag = cv2.imread(img)
      imag = Image.fromarray(np.uint8(imag))
      imag = imag.convert('L')
      img = np.asarray(imag)
      image_equalized = clahe.apply(img)
      image_equalized_img = Image.fromarray(np.uint8(image_equalized))
      image_equalized = image_equalized_img.convert('RGB')
      image_equalized = np.asarray(image_equalized)
      cv2.imwrite(ifn,image_equalized);
      imgs.append(ifn)
    return imgs

def Feature_padding(feature_maps, size):
    """
    Pad feature maps to a uniform size using bilinear interpolation.

    This function adjusts the size of each feature map in the input list to a sp

    Parameters:
    - feature_maps (list of tensors): A list of feature map tensors to be resize
    - size (tuple): The target size for the feature maps as (height, width).

    Returns:
    - list: A list of uniformly sized feature maps.
    """
    uniform_feature_maps=[]
    for feature in feature_maps:
        uniform_feature_maps.append(F.interpolate(feature, size=size, mode='bili
    return uniform_feature_maps

def multi_resolution_features(orig_images,img_size,N,clip,offset,window_size,max
    """
    Generate multi-resolution features from images using SIFT, and Random Points

    This function processes images to generate feature maps at multiple resoluti

    Parameters:
```

```
    - orig_images (list of str): List of paths to the images to be processed.
    - img_size (int): The size of the images for processing.
    - N (int): The number of keypoints to be used in SIFT.
    - clip (float): The clip limit for CLAHE.
    - max_dist (float): Maximum distance for keypoint selection in SIFT.
    - timestep (float): Timestep parameter for Diffusion Model initialization.
    - up_ft_indices (list): Indices for feature upsampling in the Diffusion Mode
    - multi_ch (bool): Flag to indicate multi-channel mode.
    - multi_img_size (int): The size of the images for multi-resolution processi
    - multi_iter (int): Number of iterations for multi-resolution processing.

    Returns:
    - tuple: A tuple of source and target feature tensors.
    """
    if multi_ch:
        src_fts,trg_fts =[],[]
        for i in range(multi_iter):
            images,pts,dft = main_initialization(orig_images,N,multi_img_size*(i
            src_ft1,trg_ft1 = dft.feature_upsampling(RetinaRegNet_Intialization(
            src_fts.append(src_ft1)
            trg_fts.append(trg_ft1)
        src_fts = Feature_padding(src_fts,(img_size,img_size))
        trg_fts = Feature_padding(trg_fts,(img_size,img_size))
        src_ft = torch.cat(src_fts, dim=1)
        trg_ft = torch.cat(trg_fts, dim=1)
    else:
        images,pts,dft = main_initialization(orig_images,N,img_size,max_dist,off
        src_ft,trg_ft = dft.feature_upsampling(RetinaRegNet_Intialization(images
    return src_ft,trg_ft

def landmarks_condition_check(orig_images, img_size, pts, t, uft, landmarks1, la
    """
    Iteratively attempts to improve image registration quality by enhancing imag
    until certain quality conditions are met or a maximum number of attempts is
    for image contrast enhancement and uses various feature transformation and s
    of landmark correspondences between two images.

    Parameters:
    - orig_images (list of str): Paths to the original images to be processed.
    - img_size (int): Size of the images to be processed, assumed to be square.
    - pts (list): List of all sampled feature keypoints in the image
    - t (float): Threshold parameter for initializing the Diffusion Model.
    - uft (float): Parameter for extracting diffusion features from the diffusio
    - landmarks1 (list of tuples): Initial landmarks as (x, y) coordinates in th
    - landmarks2 (list of tuples): Target landmarks as (x, y) coordinates in the
    - max_tries (int, optional): Maximum number of attempts to improve image reg
    - num (int, optional): Minimum required number of landmarks. Defaults to 100
    - iccl (float, optional): Inverse consistency criteria limit used in landmar
    - outlier_cond (str, optional): Condition used to determine outliers. Defaul
    - thresh (float, optional): Threshold used for filtering outliers. Defaults

    Returns:
    - tuple: Depending on the success of the registration process, this function
        The original images and the best set of landmarks found, or
        The original images and a set of default landmarks if conditions are not

    Raises:
    - AssertionError: If the number of initial and target landmarks do not match

    Notes:
```

```
        This function is particularly useful in medical imaging or computer visi
        registration is crucial for further analysis.
        The effectiveness of the registration process depends heavily on the qua
        CLAHE and other image processing techniques may not always produce the d
        are of poor quality or the initial landmarks are inaccurately defined.
    """
    imgs,lim,land_marks1, land_marks2,list_landmarks_2, list_sim_scores,list_lan
    tries, ch, = 0, 0
    assert len(landmarks1) == len(landmarks2), f"Points lengths are incompatible
    landmarks2,landmarks1 = filter_outlier_cond(landmarks2,landmarks1,outlier_cc
    list_landmarks_1.append(landmarks1)
    imgs.append(orig_images)
    list_landmarks_2.append(landmarks2)
    if len(landmarks2) < num:
        print("Image Registration Unsuccessful for Original Set of Images")
        while len(land_marks2) < num and tries< max_tries:
            print("Executing Trial", tries + 1)
            dft = DFT(orig_images, img_size, pts)
            src_ft,trg_ft = dft.feature_upsampling(RetinaRegNet_Intialization(or
            land_marks1,sim_score, land_marks2 = dft.feature_maps(src_ft,trg_ft,
            del src_ft
            del trg_ft
            torch.cuda.empty_cache()
            gc.collect()
            land_marks2,land_marks1 = filter_outlier_cond(land_marks2,land_marks
            list_landmarks_1.append(land_marks1)
            imgs.append(images)
            list_landmarks_2.append(land_marks2)
            list_sim_scores.append(np.mean(sim_score))
            tries += 1
        for i in range(len(list_landmarks_2)):
            lim.append(len(list_landmarks_2[i]))
        idx = np.argmax(np.array(lim))
        return orig_images,list_landmarks_1[idx],list_landmarks_2[idx]
    else:
        return orig_images, landmarks1, landmarks2
```

In [15]:

```
def folder_structure(path,nfn):
    """
    Creates a hierarchical folder structure for saving image registration result

    The function constructs directories for three stages of image registration r
    It ensures that the directory for each stage exists, or creates it if it doe
    to organize output from multiple stages of processing in separate folders un

    Parameters:
        - path (str): The base path where the 'Image_Registration_Results' direc
        - nfn (list of str): A list of sub-folder names to create under each sta

    Returns:
        - None: This function only creates directories and does not return any v

    Notes:
        This function uses `os.makedirs()` with `exist_ok=True` to ensure that n
        It is useful for setting up an organized structure for storing outputs f
    """
    for i in range(len(nfn)):
        os.makedirs(os.path.join(path+'_'+'Image_Registration_Results','Stage1',
        os.makedirs(os.path.join(path+'_'+'Image_Registration_Results','Stage2',
        os.makedirs(os.path.join(path+'_'+'Image_Registration_Results','Final_Re
```

```
    print("Created {} Sub Folders for saving Registration Results".format(len(nf

def images_oganization(images):
    """
    Organizes images into pairs for processing.

    This function takes a list of images and rearranges them into pairs. If the
    is even, it swaps each pair (e.g., [img1, img2] becomes [img2, img1]). If th
    it prints a warning indicating that some images do not have a pair.

    Parameters:
    - images (list of str): A list of image file paths.

    Returns:
    - imags (list of list of str): A list containing reordered pairs of image fi

    Raises:
    - UserWarning: Raises a warning if the number of images is odd, indicating n
    """
    imags = []
    if len(images) % 2 == 0:
        for i in range(0, len(images), 2):  # Loop with a step of 2
            imags.append([images[i+1], images[i]])
    else:
        print("Some Images do not have a pair")
    return imags

def sub_files_oganization(images,pnts):
    """
    Organizes images and corresponding points into three categories based on the

    This function categorizes images and their corresponding point sets into thr
    first letter of the file names ('A', 'P', 'S'). It helps in sorting images f
    pipelines or data handling strategies.

    Parameters:
    - images (list of tuple): List of tuples, each containing the file paths of
    - pnts (list): List of corresponding point data associated with each image.

    Returns:
    - tuple: Contains six lists organized into three groups:
        - imags_A, pnts_A: Lists containing images and points starting with 'A'.
        - imags_P, pnts_P: Lists containing images and points starting with 'P'.
        - imags_S, pnts_S: Lists containing images and points starting with 'S'.

    Note:
      It is assumed that the image filenames are structured in a way that their
      from the first letter of the basename of the file path.
    """
    imags_A ,imags_P,imags_S = [],[],[]
    pnts_A ,pnts_P,pnts_S = [],[],[]
    for i in range(len(images)):
        if os.path.splitext(os.path.basename(images[i][0]))[0][:1] =='A':
            imags_A.append(images[i])
            pnts_A.append(pnts[i])
        elif os.path.splitext(os.path.basename(images[i][0]))[0][:1] =='P':
            imags_P.append(images[i])
            pnts_P.append(pnts[i])
        else:
            imags_S.append(images[i])
```

```
            pnts_S.append(pnts[i])
    return imags_A,pnts_A,imags_P,pnts_P,imags_S,pnts_S


def text_points_parser(pnts):
    """
    Extract point coordinates from a text file.

    Parameters:
    - pnts (str): Path to the text file containing point coordinates.

    Returns:
    - tuple: Lists of fixed points and moving points.
    """
    fixed_pnts = []
    moving_pnts = []
    with open(pnts, 'r') as file:
        for line in file:
            points = [float(coord) for coord in line.strip().split()]
            fps = tuple(points[:2])
            lps = tuple(points[2:])
            fixed_pnts.append(fps)
            moving_pnts.append(lps)
    return fixed_pnts,moving_pnts

def coordinates_rescaling_high_scale(pnts,H,W,img_shape):
    """
    Rescale a list of coordinates based on given height and width ratios.

    Parameters:
    - pnts (list of tuples): List of (x, y) coordinates to be rescaled.
    - H (int): Original height.
    - W (int): Original width.
    - img_shape (int): Desired image dimension (assumes square shape).

    Returns:
    - list of tuples: List of rescaled (x, y) coordinates.
    """
    scaled_points=[]
    for row in pnts:
        a = (row[0]/W)*img_shape[1]
        b = (row[1]/H)*img_shape[0]
        scaled_points.append((a,b))
    return scaled_points


def coordinates_rescaling(pnts,H,W,img_shape):
    """
    Rescale a list of coordinates based on given height and width ratios.

    Parameters:
    - pnts (list of tuples): List of (x, y) coordinates to be rescaled.
    - H (int): Original height.
    - W (int): Original width.
    - img_shape (int): Desired image dimension (assumes square shape).

    Returns:
    - list of tuples: List of rescaled (x, y) coordinates.
    """
    scaled_points=[]
```

```
    for row in pnts:
        a = (row[0]/W)*img_shape
        b = (row[1]/H)*img_shape
        scaled_points.append((a,b))
    return scaled_points

def coordinates_processing(image1,image2,fpnts,mpnts,img_shape=256):
    """
    Process and rescale coordinates for two images.

    Parameters:
    - image1 (str): Path to the first image.
    - image2 (str): Path to the second image.
    - fpnts (list of tuples): List of (x, y) coordinates related to the first im
    - mpnts (list of tuples): List of (x, y) coordinates related to the second i
    - img_shape (int, optional): Desired image dimension for rescaling. Default

    Returns:
    - tuple: A tuple containing:
        - Tuple: Height and Width of the second image.
        - Tuple: Height and Width of the first image.
        - int: Maximum of the heights and widths of both images.
        - list of tuples: Scaled coordinates for the first image.
        - list of tuples: Scaled coordinates for the second image.
        - list of tuples: Scaled original coordinates for the second image.
    """
    H1,W1,C1 = cv2.imread(image1).shape
    H2,W2,C2 = cv2.imread(image2).shape
    scaled_moving_points = coordinates_rescaling(mpnts,H1,W1,img_shape)
    scaled_fixed_points = coordinates_rescaling(fpnts,H2,W2,img_shape)
    scaled_original_moving_points = coordinates_rescaling(mpnts,H1,W1,max(max(H1
    return (H2,W2),(H1,W1),max(max(H1,W1),max(H2,W2)),scaled_fixed_points,scaled

def feature_scaling(images,fixed_points,moving_points,img_shape):
    """
    Apply feature scaling to given images and their associated points.

    Parameters:
    - images (list): List of tuples containing image paths for fixed and moving
    - fixed_points (list): List of fixed points corresponding to each image.
    - moving_points (list): List of moving points corresponding to each image.
    - img_shape (int): Desired image dimension for rescaling.

    Returns:
    - tuple: A tuple containing:
        - list: Sizes of fixed images.
        - list: Sizes of moving images.
        - list: Maximum of the heights and widths of the images.
        - list: Fixed points after scaling.
        - list: Moving points after scaling.
        - list: Scaled moving points.
    """
    fixed_image_size,moving_image_size,max_image_size,fixed_pointss,moving_point
    for i in range(len(images)):
        fhs,mhss,mhs,fpnts,mpnts,scmpnts = coordinates_processing(images[i][0],i
        fixed_image_size.append(fhs)
        moving_image_size.append(mhss)
        max_image_size.append(mhs)
        fixed_pointss.append(fpnts)
        moving_pointss.append(mpnts)
```

```
        scaled_moving_points.append(scmpnts)
    return fixed_image_size,moving_image_size,max_image_size,fixed_pointss,movin

def text_file_processing(cnts):
    """
    Processes a list of text files containing point data to extract fixed and mo

    Parameters:
    - cnts (list of str): A list of file paths where each file contains coordina

    Returns:
    - tuple of lists: A tuple containing two lists:
        - fixed_points (list): A list of fixed points extracted from the text fi
        - moving_points (list): A list of moving points extracted from the text

    Notes:
      This function skips files named '.ipynb_checkpoints' and only processes va
      Each file is expected to have fixed and moving points in a specific format
    """
    fixed_points,moving_points =[],[]
    for i in cnts:
        if os.path.isfile(i) and i != '.ipynb_checkpoints':
            fps,mps = text_points_parser(i)
            fixed_points.append(fps)
            moving_points.append(mps)
        else:
            continue
    return fixed_points,moving_points

def data_organization(pth,img_shape=256,files =['Images','Ground Truth']):
    """
    Organizes and processes image and point data from specified directories.

    Parameters:
    - pth (str): The base directory path that contains the 'Images' and 'Ground
    - img_shape (int, optional): The target size for resizing the images. Defaul
    - files (list of str, optional): The list of directory names to process. Def

    Returns:
    - tuple: Contains a large number of elements, including organized lists of i
             and metadata about image and point processing.

    Notes:
      This function segregates images and point data based on their filename ini
      such as 'A', 'P', 'S' for different processing tasks.
      Each category of files is further processed to extract and scale point dat
      The function uses several other functions such as `folder_structure`, `ima
      `sub_files_organization` to structure and organize data.
    """
    images , cnts ,fns,fnn  =[], [], [] ,[]
    for i in files:
        for j in sorted(os.listdir(os.path.join(pth,str(i)))):
            if i == 'Images':
                images.append(os.path.join(pth,str(i),j))
                fns.append(j[0])
            else:
                cnts.append(os.path.join(pth,str(i),j))
    images = [img for img in images if not img.startswith('.')]
    cnts = [pnt for pnt in cnts if not pnt.startswith('.')]
    folder_structure(pth,np.unique(fns))
```

```
    images = images_oganization(images)
    images_A,cnts_A,images_P,cnts_P,images_S,cnts_S = sub_files_oganization(imag
    fixed_points , moving_points = text_file_processing(cnts)
    fixed_points_A , moving_points_A = text_file_processing(cnts_A)
    fixed_points_P , moving_points_P = text_file_processing(cnts_P)
    fixed_points_S , moving_points_S = text_file_processing(cnts_S)
    fixed_image_size,moving_image_size,max_image_size,fixed_pointss,moving_point
    fixed_image_size_A,moving_image_size_A,max_image_size_A,fixed_pointss_A,movi
    fixed_image_size_P,moving_image_size_P,max_image_size_P,fixed_pointss_P,movi
    fixed_image_size_S,moving_image_size_S,max_image_size_S,fixed_pointss_S,movi
    return images,images_A,images_P,images_S,fixed_image_size,fixed_image_size_A
```

In [16]:

```
def RetinaRegNet_Intialization(filelist,img_size = 256,timestep = 75,up_ft_index
    """
    Initialize RetinaRegNet by processing a list of image files.

    Parameters:
    - filelist (list of str): List of paths to image files for feature extractio
    - img_size (int, optional): Desired size for resizing images. Default is 256
    - timestep (int, optional): Time step for the intializing the diffusion mode
    - up_ft_index (int, optional): Index for the extracting diffusion features f

    Returns:
    - ft (torch.Tensor): A tensor containing the Diffusion features of the image

    Notes:
        The function uses the SDFeaturizer from the 'stabilityai/stable-diffusio
        from each image. After processing all images, the extracted features are
        To avoid memory issues, the function cleans up resources after processin
    """
    ft = []
    imglist = []
    dfm = SDFeaturizer(sd_id='stabilityai/stable-diffusion-2-1')
    for filename in filelist:
        img = Image.open(filename).convert('RGB')
        img = img.resize((img_size, img_size))
        imglist.append(img)
        img_tensor = (PILToTensor()(img) / 255.0 - 0.5) * 2
        ft.append(dfm.forward(img_tensor,
                              timestep,
                              up_ft_index,
                              prompt='FIRE',
                              ensemble_size=8))
    ft = torch.cat(ft, dim=0)

    del dfm
    torch.cuda.empty_cache()
    gc.collect()
    return ft
```

In [17]:

```
def main(orig_images,rpth,ifn,stage_num,img_size=256,up_ft_indices = 1,timestep
    """
    Perform image registration and point correspondence using a series of proces

    Parameters:
    - images (list): A list of input images for registration.
    - rpth (str): Path to save the resulting registered images.
    - ifn (str): File name prefix for the saved images.
    - stage_num (int): Stage number for referencing in plots and outputs.
```

```
    - img_size (int, optional): Size of the input images (default is 256).
    - up_ft_indices (int, optional): Up-sampling factor for feature indices (def
    - timestep (int, optional): Time step for feature extraction (default is 75)
    - N (int, optional): Number of keypoints to extract (default is 50).
    - offset (float, optional): Offset parameter for feature extraction (default
    - window_size (int, optional): Size of the window for feature extraction (de
    - max_dist (int, optional): Maximum distance for feature matching (default i
    - iccl (int, optional): ICC level for feature matching (default is 3).
    - outlier_cond (str, optional): Condition for outlier removal (default is 'a
    - thresh (int, optional): Threshold value for outlier removal (default is 20
    - max_tries (int, optional): Maximum number of attempts for matching feature
    - num (int, optional): Number of iterations for matching features (default i
    - clip (float, optional): Clip parameter for image enhancement (default is 1
    - disp_clip (float, optional): Clip parameter for enhancing quality of image
    - multi_ch (bool, optional): Flag indicating whether to use multi-channel pr
    - multi_iter (int, optional): Number of iterations for multi-channel process
    - multi_img_size (int, optional): Size of images for multi-channel processin

    Returns:
    - original (list): List of original image points.
    - computed (list): List of computed image points after registration.

    Note:
        This function performs various processing steps including feature extrac
        outlier removal, and image registration.
        It saves the resulting registered images in the specified directory.
        If the image registration is unsuccessful, empty lists are returned for
    """
    images,pts,dft = main_initialization(orig_images,N,img_size,max_dist,offset,
    src_ft,trg_ft = multi_resolution_features(orig_images,img_size,N,clip,offset
    pnts,rmaxs, rspts = dft.feature_maps(src_ft,trg_ft,iccl)
    del src_ft
    del trg_ft
    torch.cuda.empty_cache()
    gc.collect()
    images,original,computed = landmarks_condition_check(images, img_size, pts,
    if len(computed)!=0:
        image_point_correspondences(images[::-1],img_size,computed,original,rpth
        return original,computed
    else:
        print("Image Registration is Unsuccessful for the presented Images due t
        return [],[]
    torch.cuda.empty_cache()
```

In [18]:
```
img_size = 920
images,images_A,images_P,images_S,fixed_image_size,fixed_image_size_A,fixed_imag
```

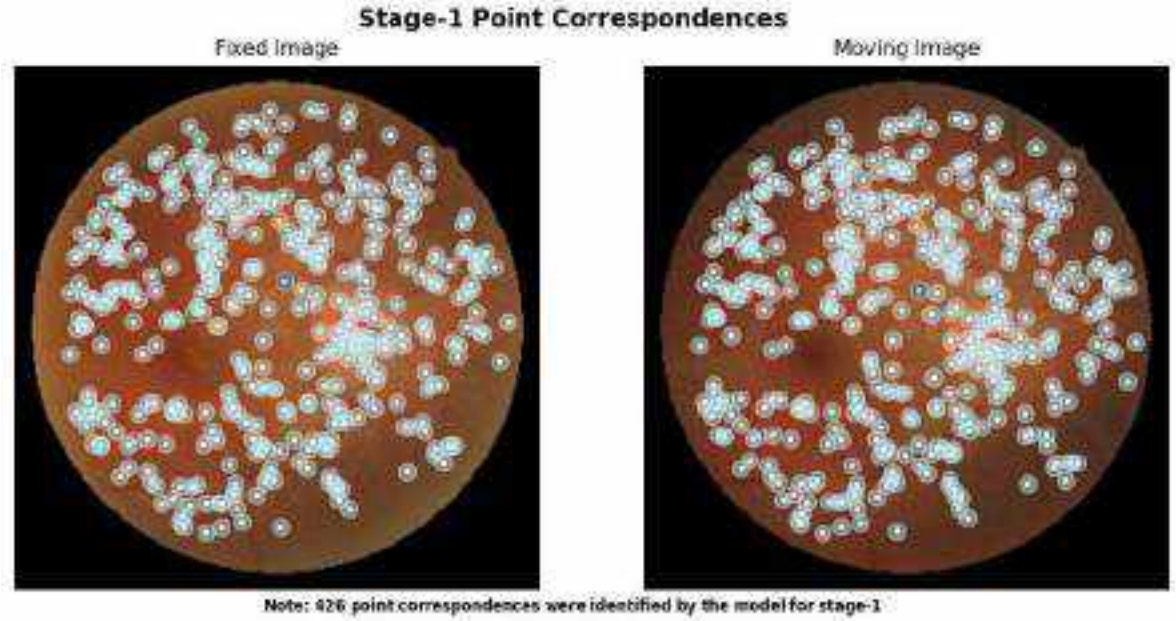
```
Created 3 Sub Folders for saving Registration Results
```

Class-A

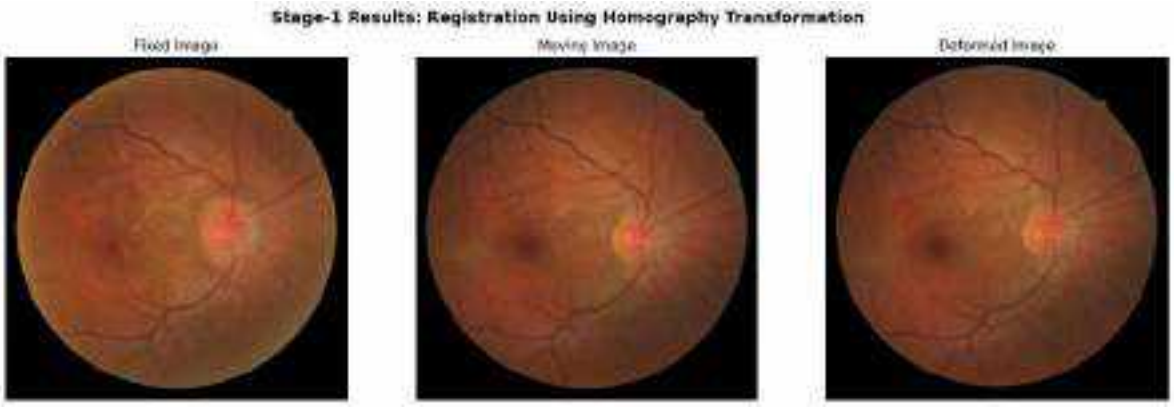
In [19]:
```
landmark_errors1=[]
for i in range(len(images_A)):
    print("Case {}".format(i))
    print("Loading Fixed Images {0} Moving Image{1} to the framework".format(ima
    original_low_res,computed_low_res = main(images_A[i],os.path.join(os.getcwd(
    imags,imgs,homography_matrix_low_res = compute_homography_matrix_and_plot(im
    if len(homography_matrix_low_res) !=0:
        transformed_points_hom = transform_points_homography(scaled_moving_point
```

```
        transformed_points_high_res_hom =  coordinates_rescaling(transformed_poi
        original_low_res,computed_low_res = main(imags,os.path.join(os.getcwd(),
        imag,imgs,polynomial_matrix_low_res = compute_third_order_polynomial_mat
        if len(polynomial_matrix_low_res) !=0:
            ## rescaled version for dispaly purposes
            transformed_points_poly = transform_points_third_order_polynomial(tr
            original_image_point_correspondences(imag,images_A[i][0],img_size, s
            ### Original Version for computation of errors
            polynomial_matrix = transform_points_third_order_polynomial_matrix(c
            bef_error = compute_landmark_error(fixed_points_A[i],fixed_image_siz
            aft_error = compute_landmark_error_fixed_space(polynomial_matrix,fix
            print("Mean Landmark Error for Case {0} Before Registration is {1} p
            print("Mean Landmark Error for Case {0} After Registration is {1} pi
            landmark_errors1.append(aft_error)
        else:
            landmark_errors1.append(10000)
    else:
        landmark_errors1.append(10000)
```

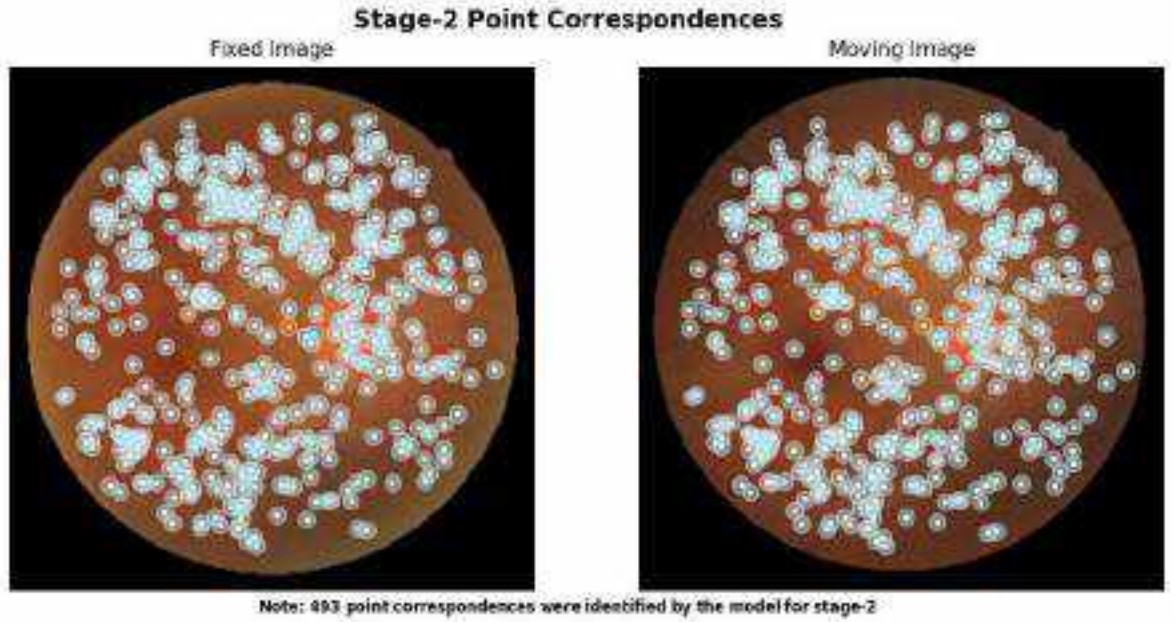
```
Case 0
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A01_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A01_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
/blue/weishao/vi.sivaraman/conda/envs/VBS_MRC/lib/python3.11/site-packages/torch/
nn/modules/conv.py:459: UserWarning: Applied workaround for CuDNN issue, install
nvrtc.so (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:80.)
  return F.conv2d(input, weight, bias, self.stride,
/scratch/local/29752099/ipykernel_2239179/635223464.py:100: UserWarning: To copy
construct from a tensor, it is recommended to use sourceTensor.clone().detach() o
r sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(so
urceTensor).
  points_indices = torch.tensor(pts_list)
```



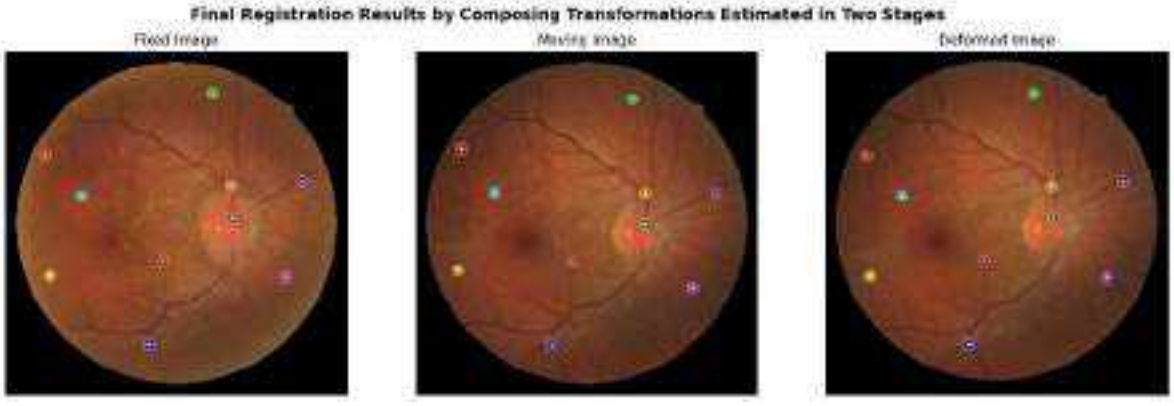

```
Homography Matrix:
[[ 9.93176661e-01  6.52462074e-02 -3.26397629e+01]
 [-7.02771972e-02  9.94469971e-01  2.50047621e+01]
 [-6.23491741e-06 -4.78400056e-06  1.00000000e+00]]
```



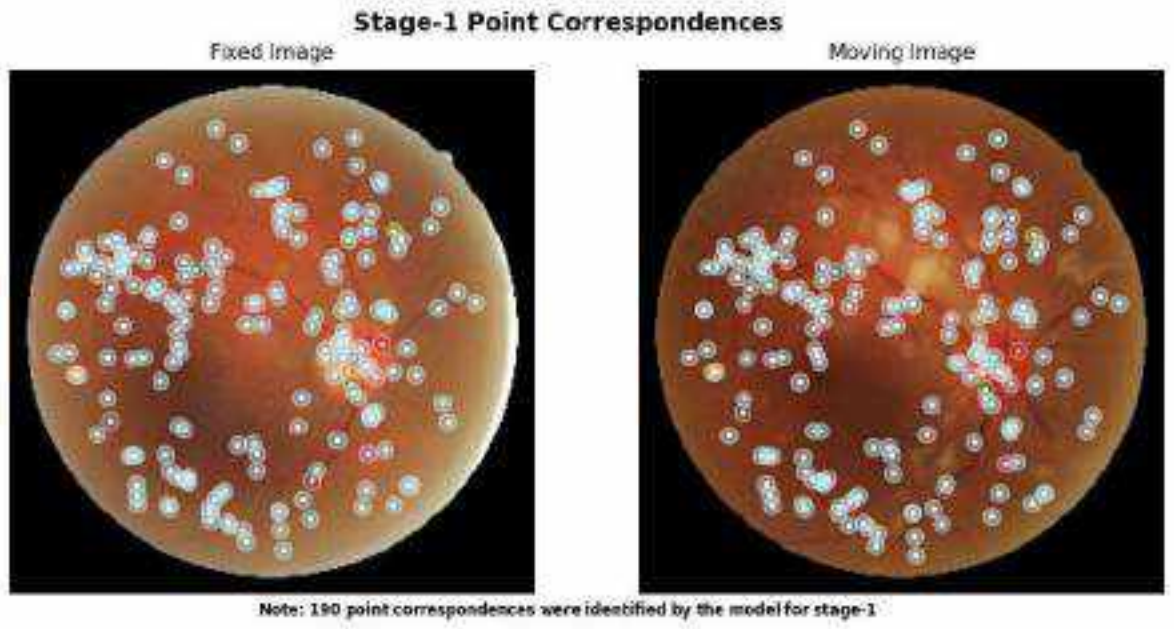

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 493 point correspondences were identified by the model for stage-2

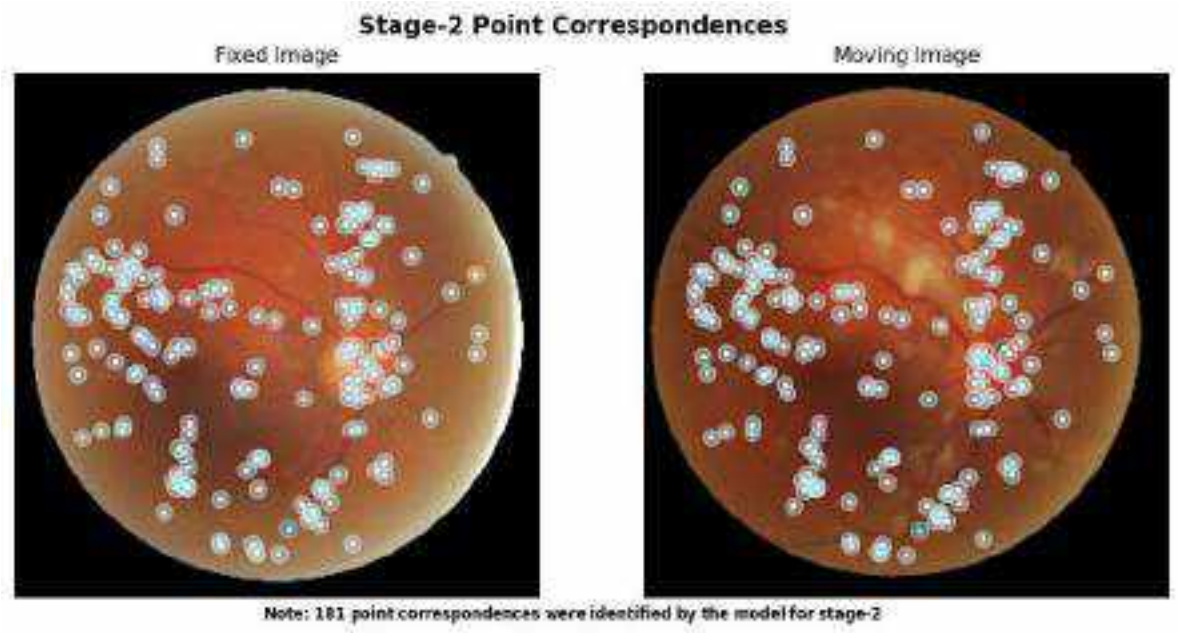


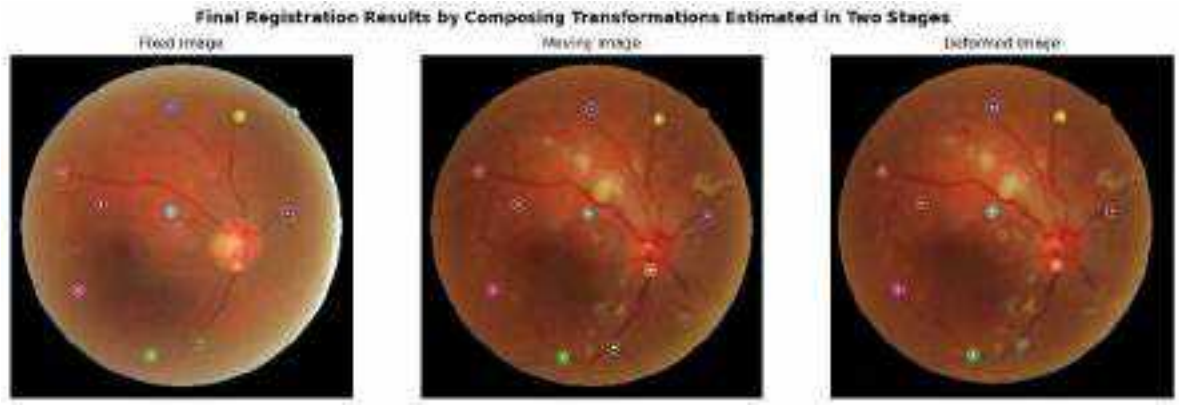
```
Mean Landmark Error for Case 0 Before Registration is 65.88031817630566 pixels
Mean Landmark Error for Case 0 After Registration is 3.467383267429482 pixels
Case 1
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A02_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A02_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



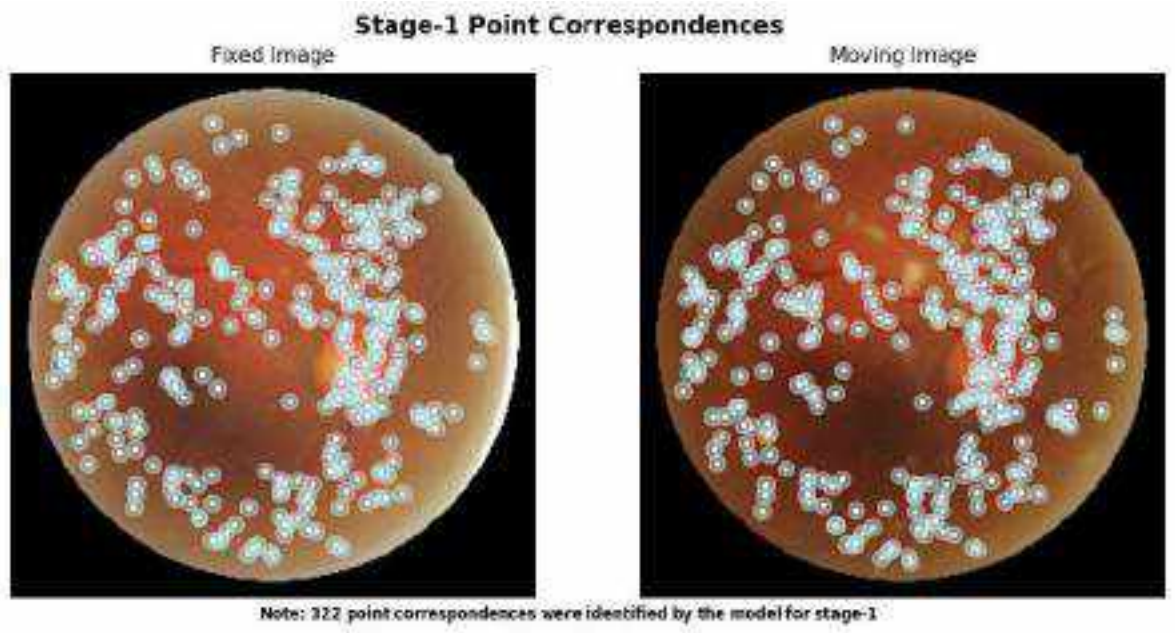

```
Homography Matrix:
[[ 1.01667190e+00  1.79133217e-02 -2.62203968e+01]
 [-1.05018814e-02  9.92957998e-01  5.10924109e+00]
 [ 3.11532848e-05 -1.32977181e-05  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

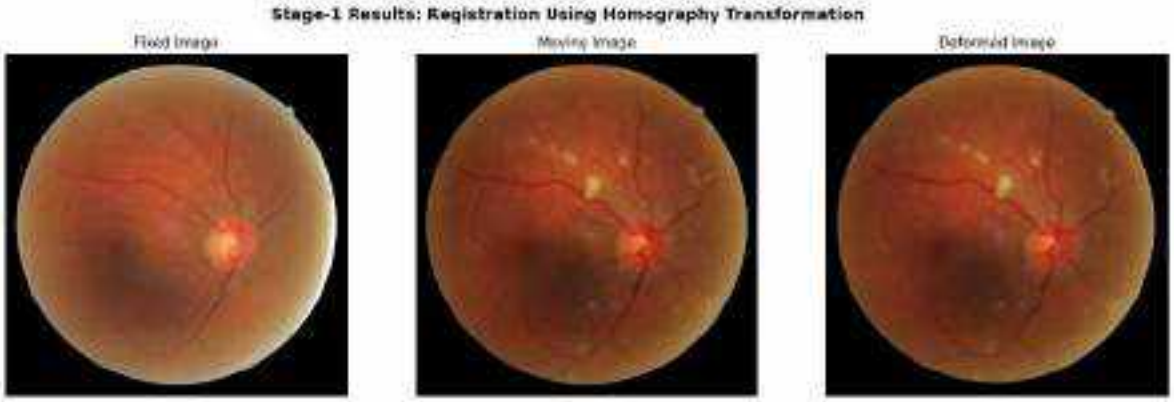




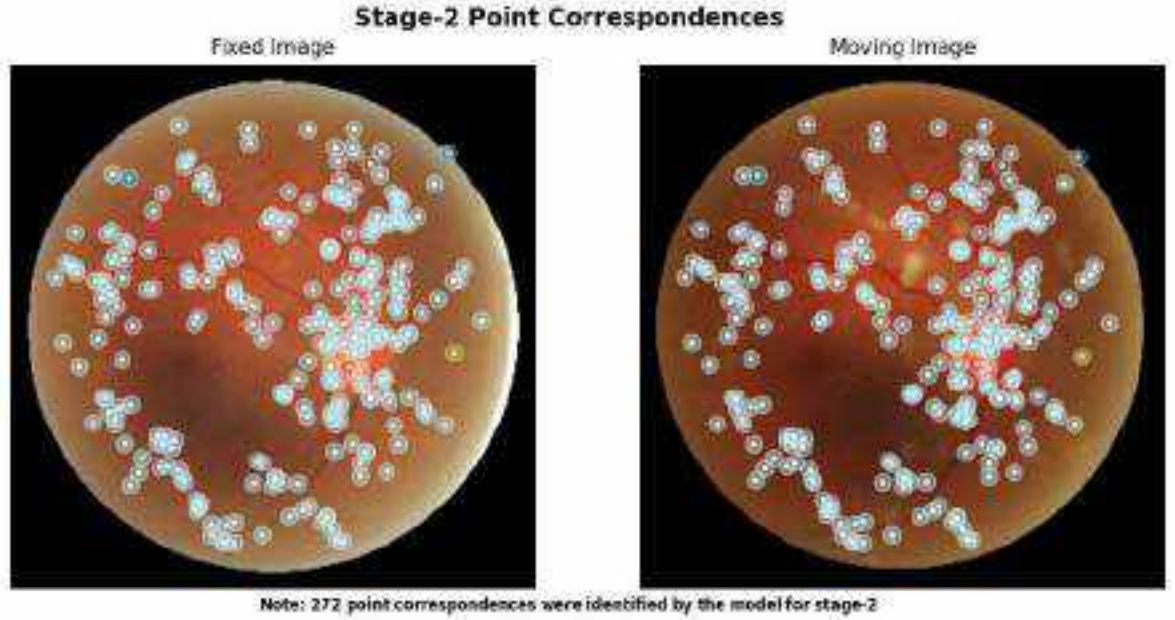

```
Mean Landmark Error for Case 1 Before Registration is 54.6511193864919B5 pixels
Mean Landmark Error for Case 1 After Registration is 8.728854133923777 pixels
Case 2
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A03_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A03_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

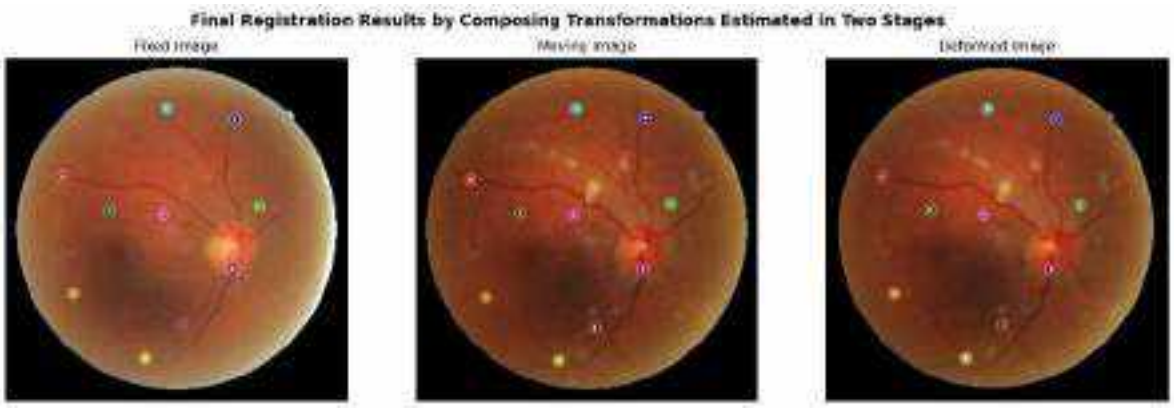



```
Homography Matrix:
[[ 9.99057813e-01 -2.71850593e-02  1.13989962e+01]
 [ 2.63727583e-02  9.88481082e-01 -9.82874451e+00]
 [ 1.15767569e-05 -9.94386972e-06  1.00000000e+00]]
```

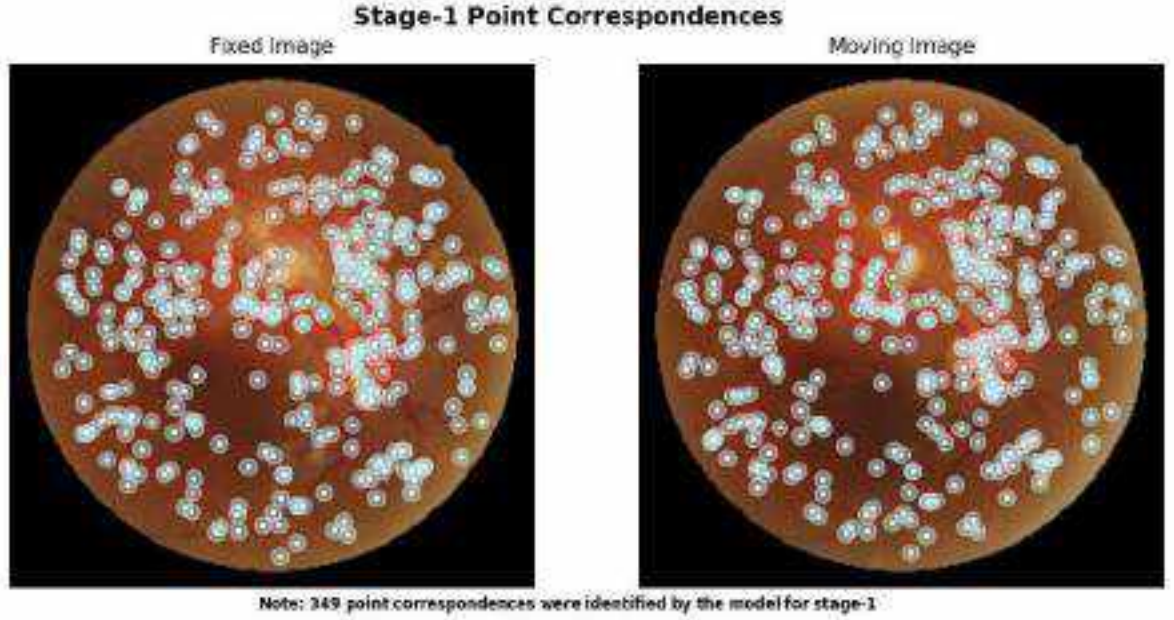



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

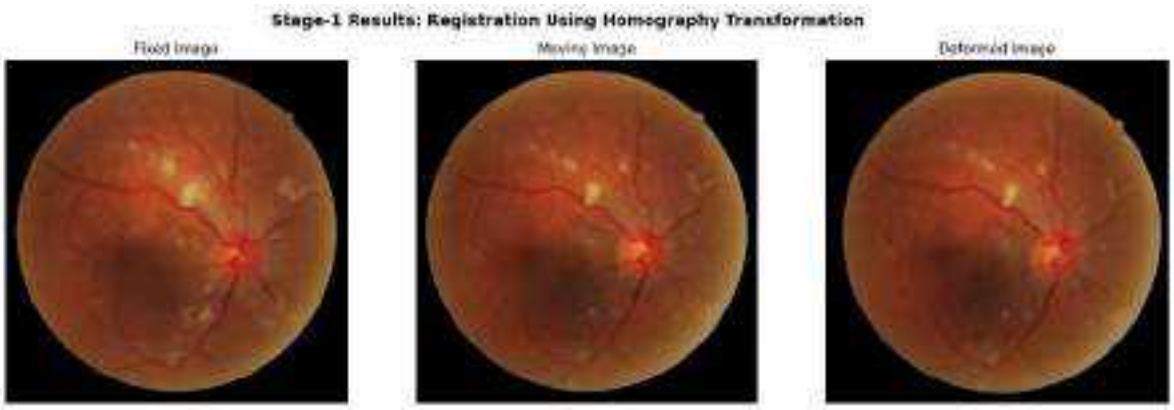




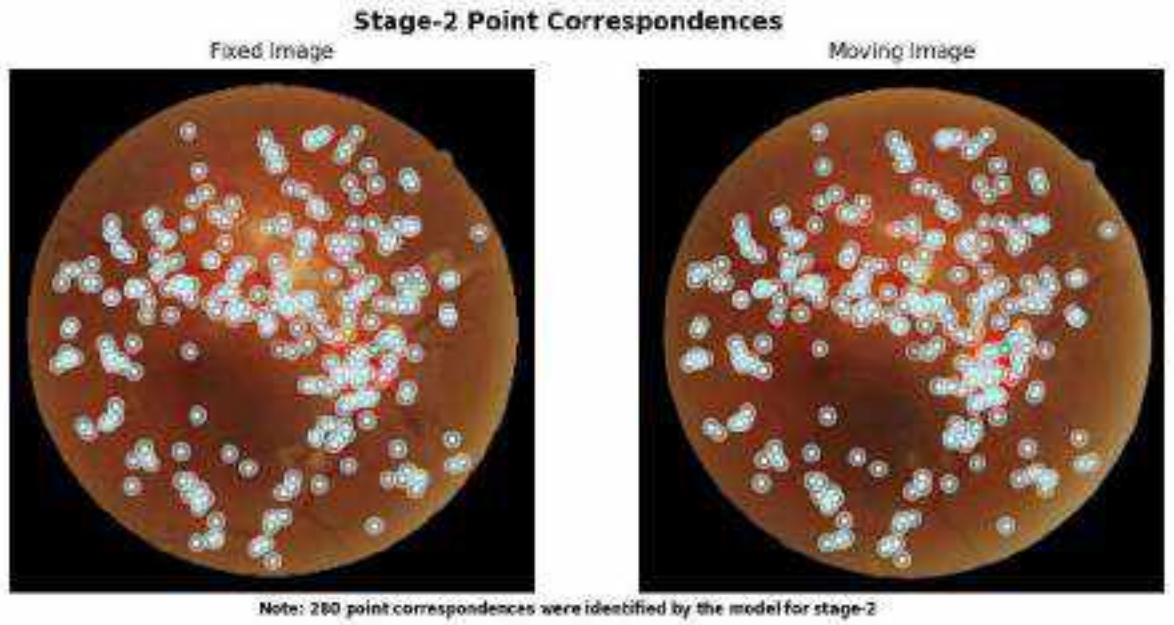

```
Mean Landmark Error for Case 2 Before Registration is 20.328151948111724 pixels
Mean Landmark Error for Case 2 After Registration is 4.313325772933447 pixels
Case 3
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A04_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A04_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



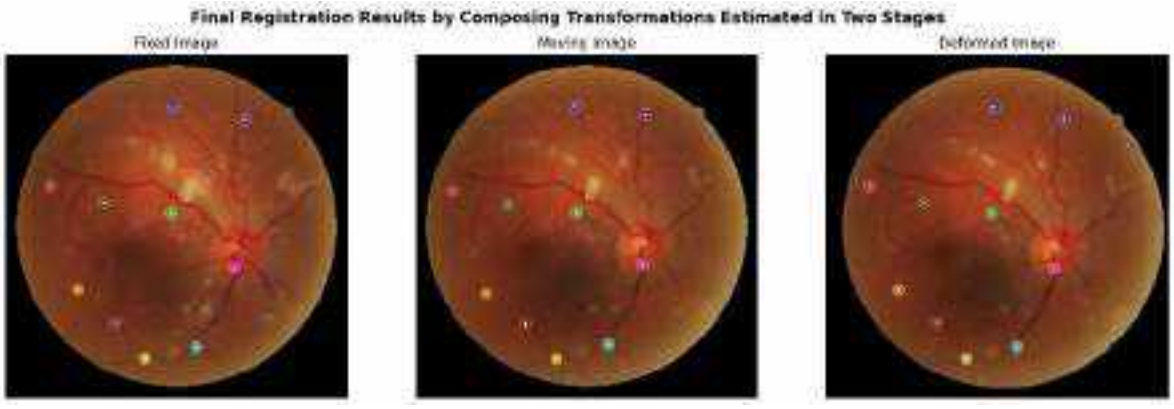

```
Homography Matrix:
[[ 9.94154739e-01 -4.30583599e-02  3.44676196e+01]
 [ 4.39179468e-02  9.97147827e-01 -1.52203557e+01]
 [ 9.32918039e-07 -1.33136641e-06  1.00000000e+00]]
```



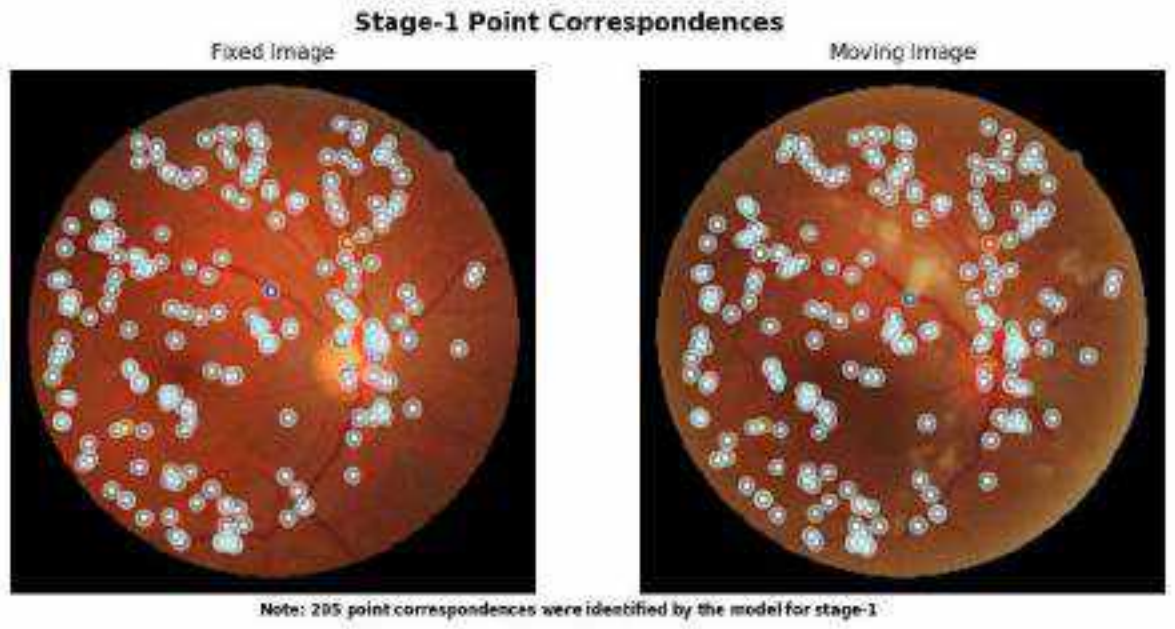

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



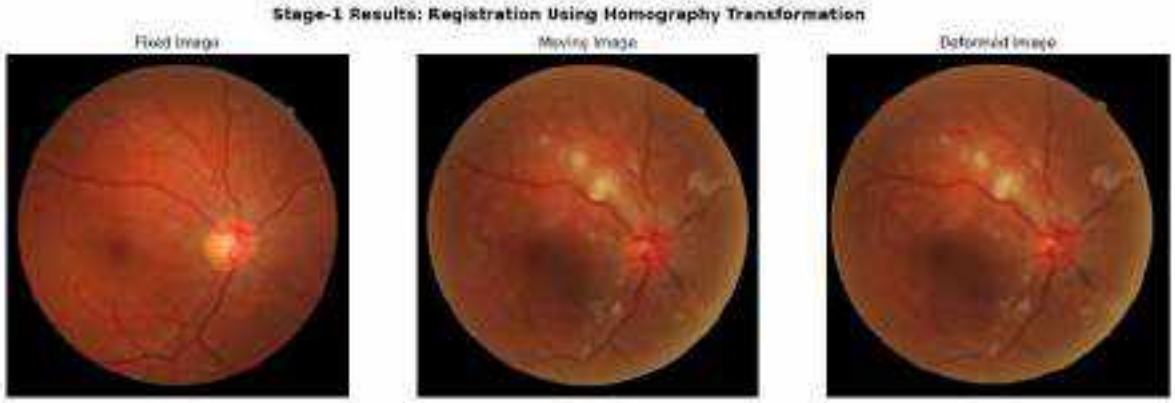
Note: 280 point correspondences were identified by the model for stage-2



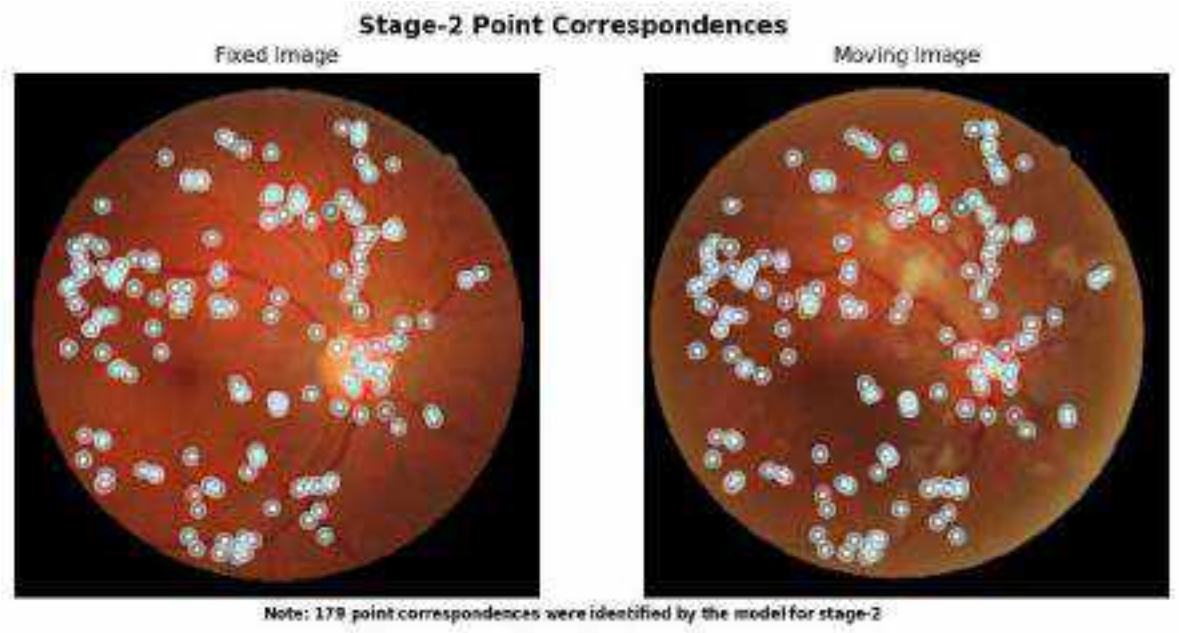
```
Mean Landmark Error for Case 3 Before Registration is 46.820895914362744 pixels
Mean Landmark Error for Case 3 After Registration is 4.736436797429592 pixels
Case 4
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A05_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A05_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

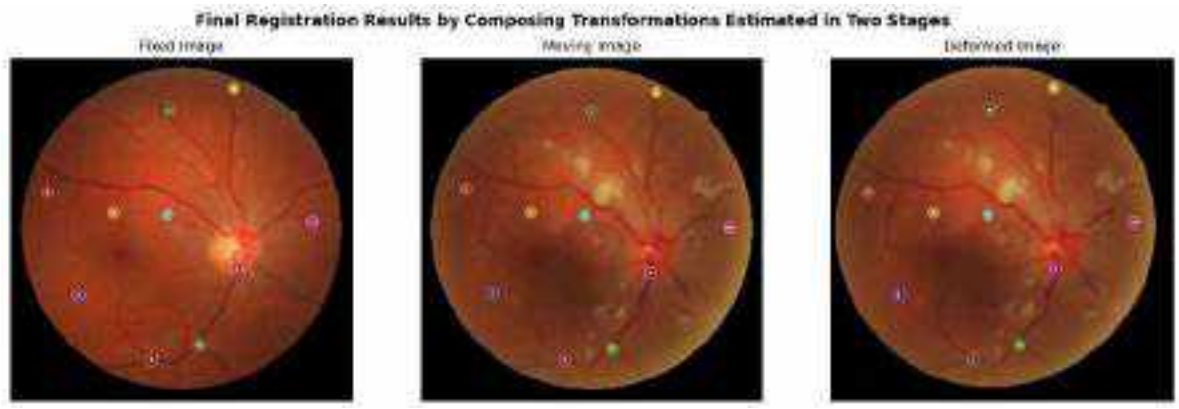



```
Homography Matrix:
[[ 9.98014140e-01  3.57349282e-02 -3.59258008e+01]
 [-3.92416500e-02  9.98524038e-01  1.62970670e+01]
 [-6.68727009e-06 -3.29287600e-06  1.00000000e+00]]
```

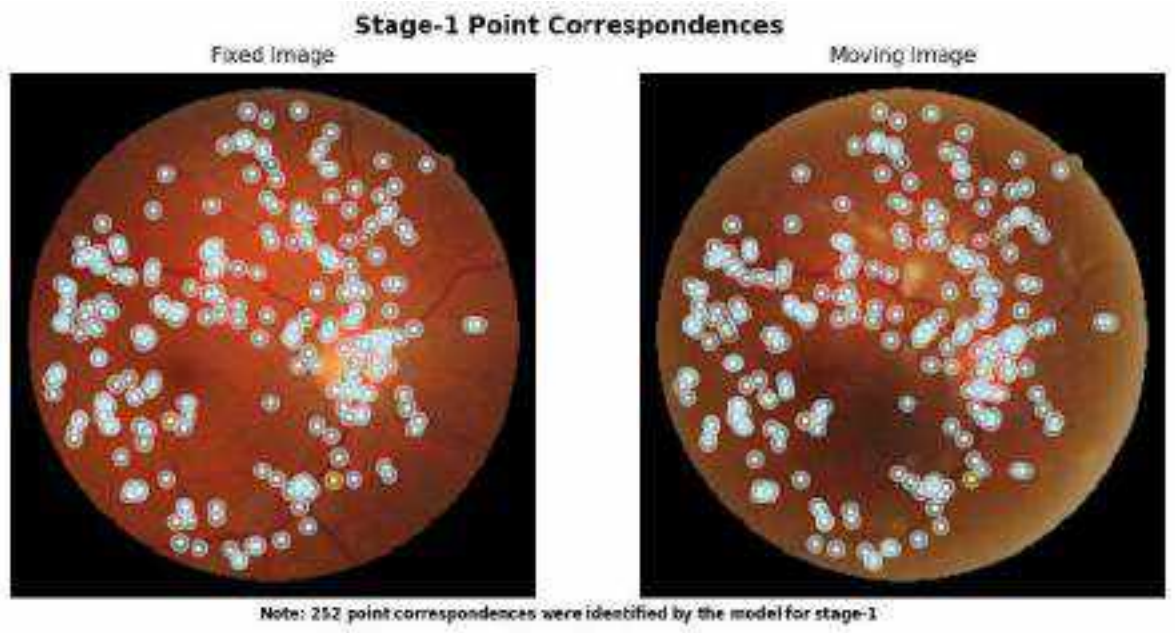



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



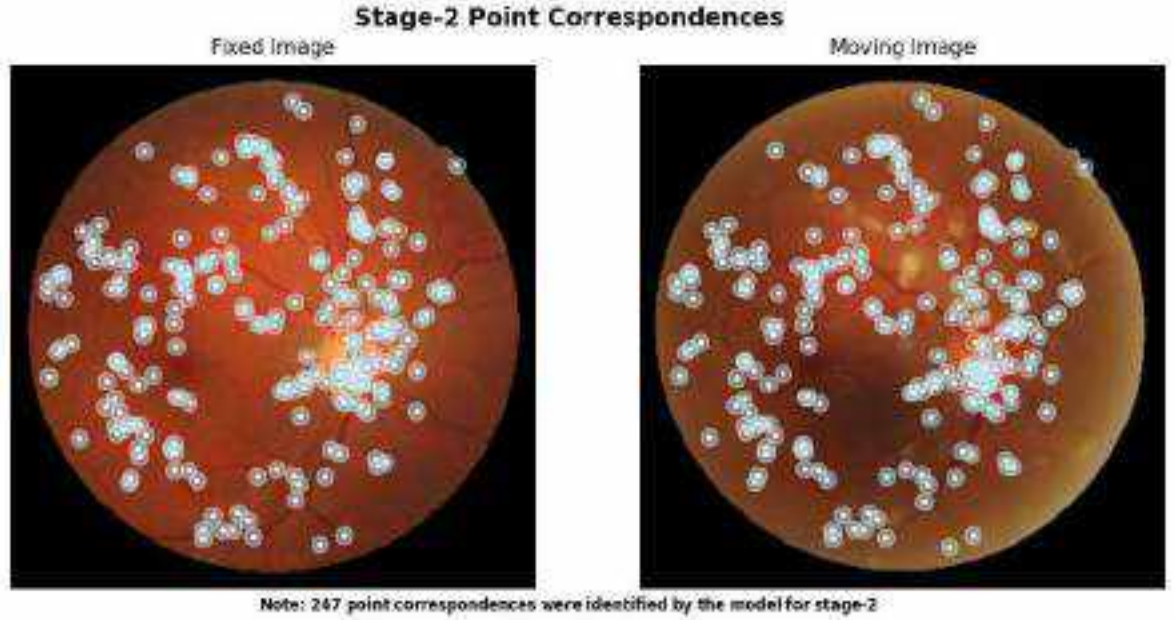


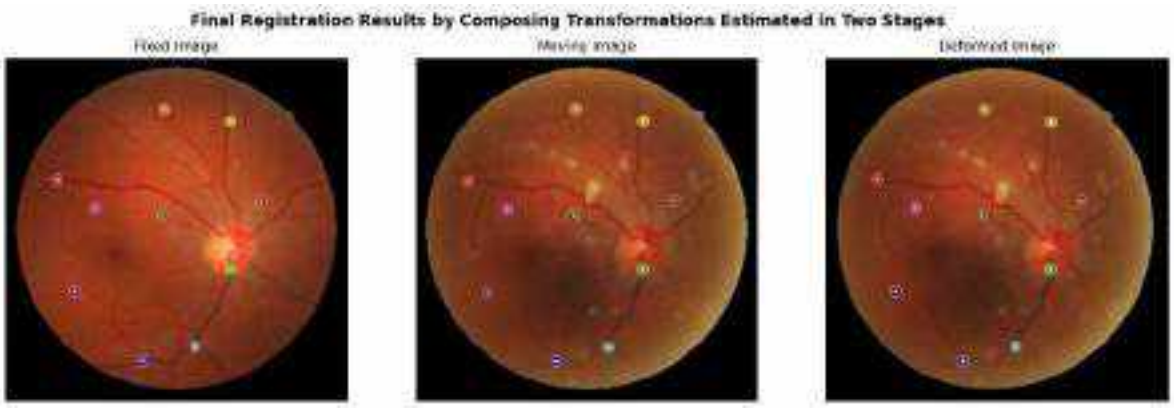

```
Mean Landmark Error for Case 4 Before Registration is 61.85317113213364 pixels
Mean Landmark Error for Case 4 After Registration is 5.744320855334431 pixels
Case 5
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A06_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A06_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



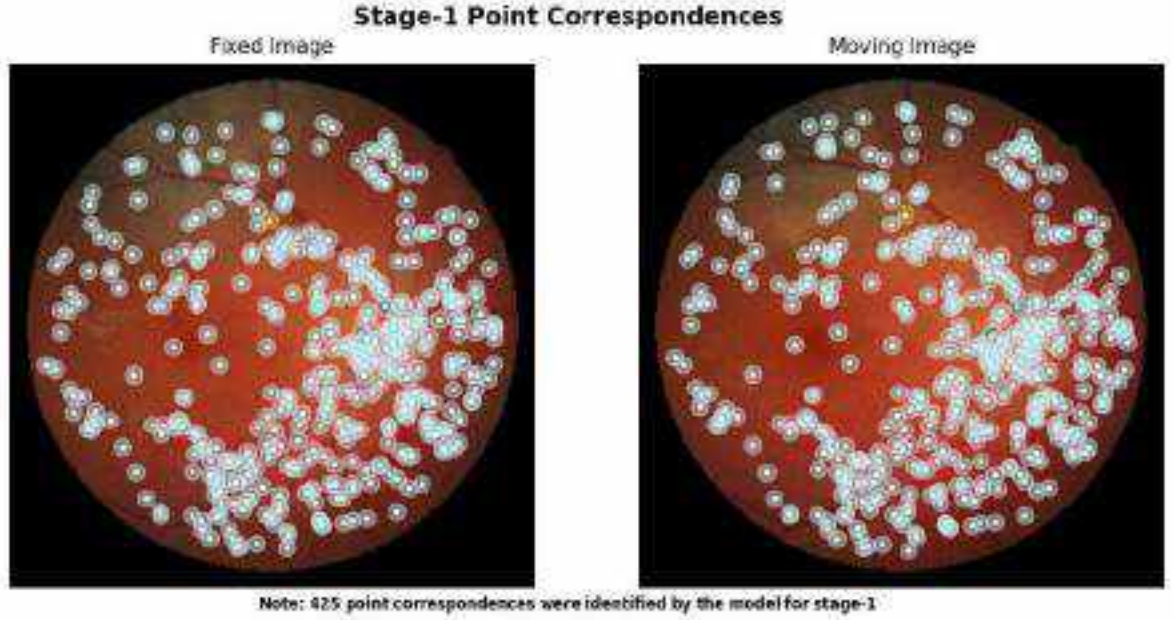

```
Homography Matrix:
[[ 9.84772653e-01 -7.32944235e-03  8.85633558e-01]
 [ 5.98626227e-03  9.88067675e-01 -1.75876976e+00]
 [-1.32374039e-05 -3.79651492e-06  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

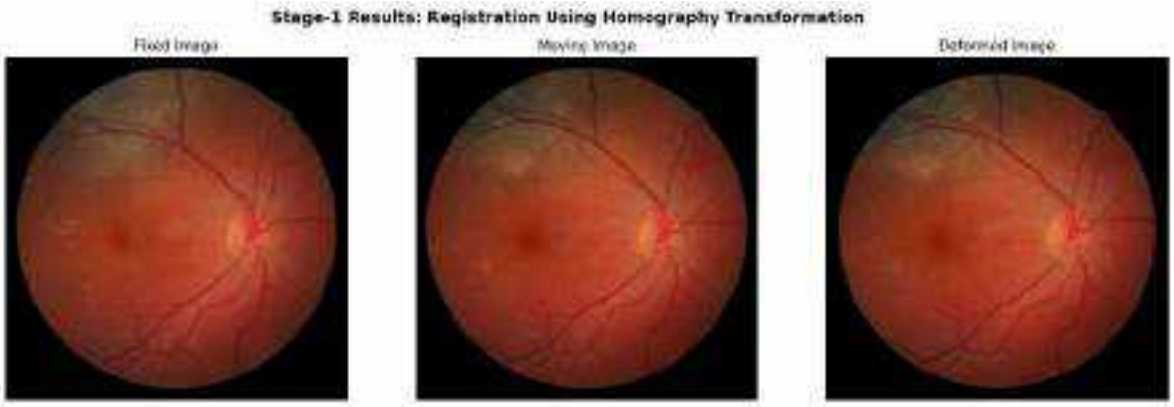




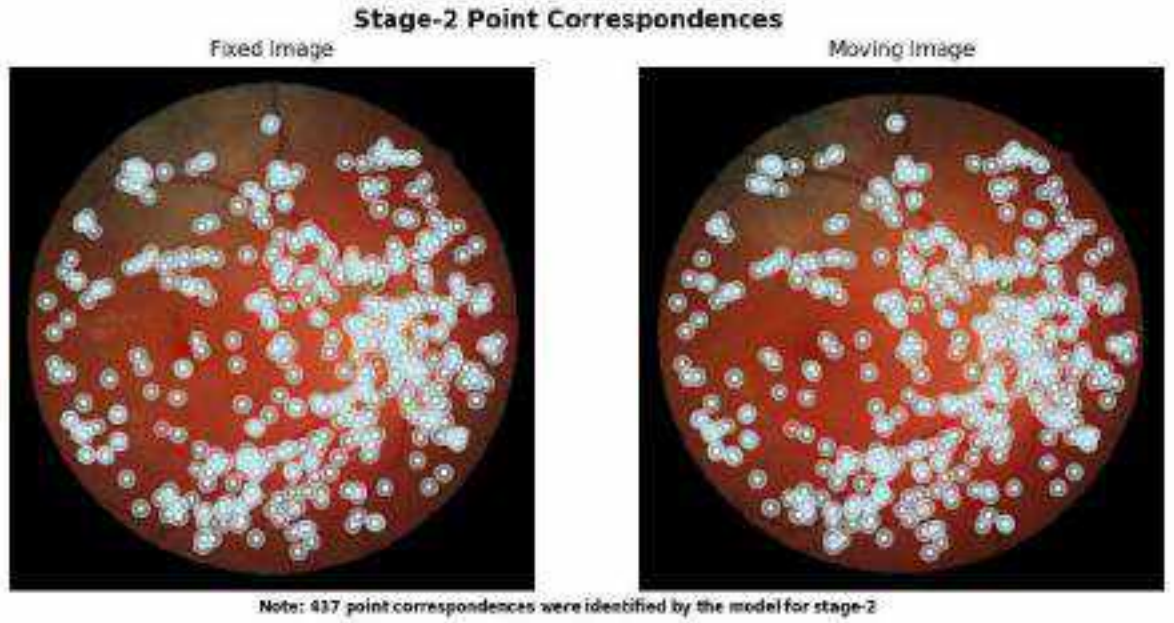

```
Mean Landmark Error for Case 5 Before Registration is 18.097080152464947 pixels
Mean Landmark Error for Case 5 After Registration is 4.521428464701847 pixels
Case 6
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A07_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A07_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



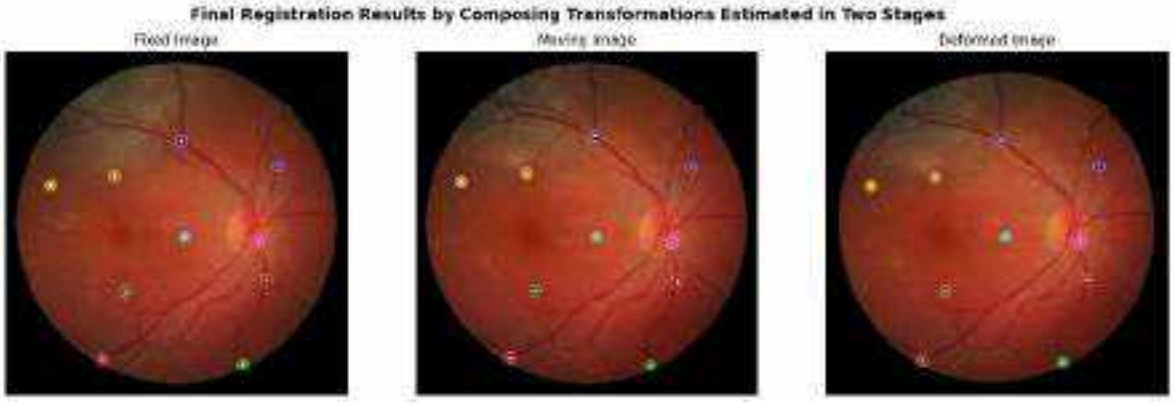

```
Homography Matrix:
[[ 9.55857959e-01  9.15575604e-03  2.02463383e-01]
 [-3.30429886e-02  9.37344006e-01  3.51518850e+01]
 [-2.16488816e-05 -3.21989254e-05  1.00000000e+00]]
```



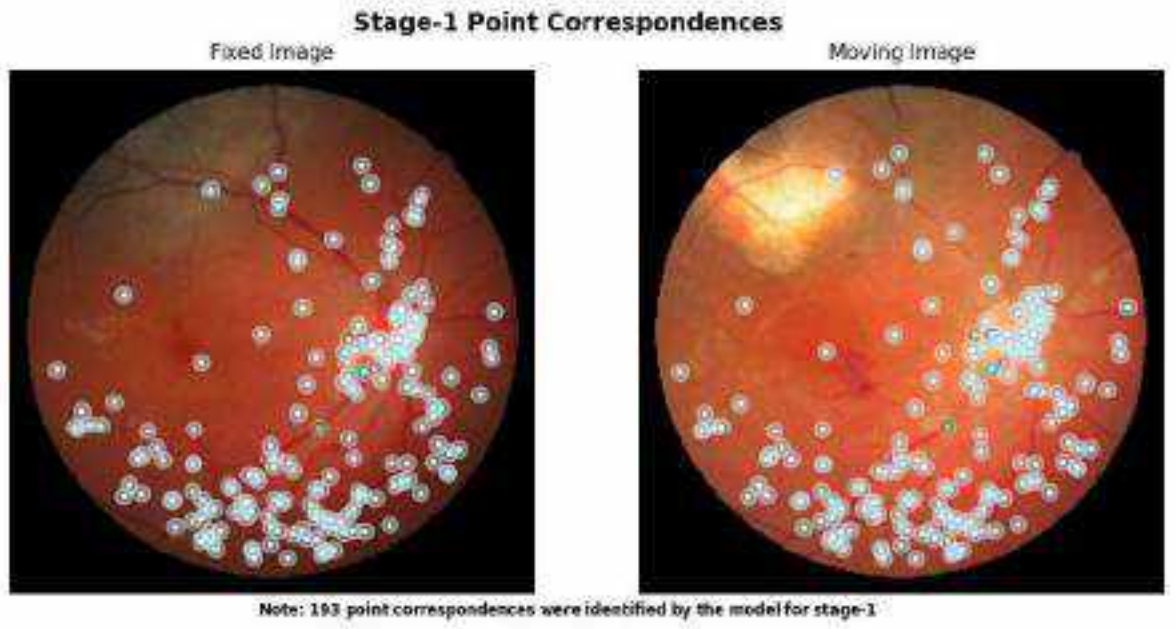

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



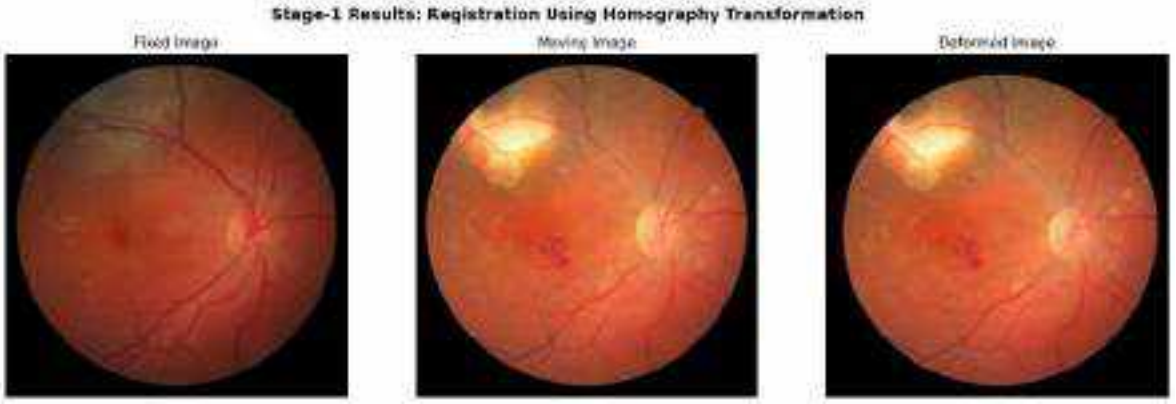
Note: 417 point correspondences were identified by the model for stage-2



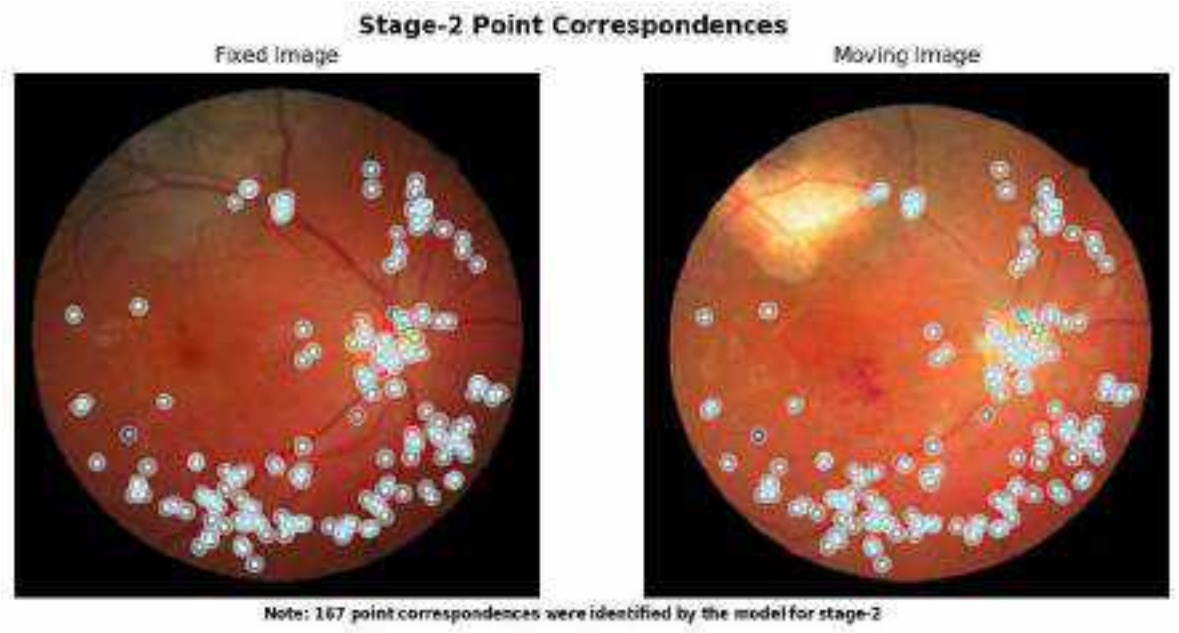
```
Mean Landmark Error for Case 6 Before Registration is 25.24155690814927 pixels
Mean Landmark Error for Case 6 After Registration is 7.45208949617111 pixels
Case 7
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A08_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A08_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

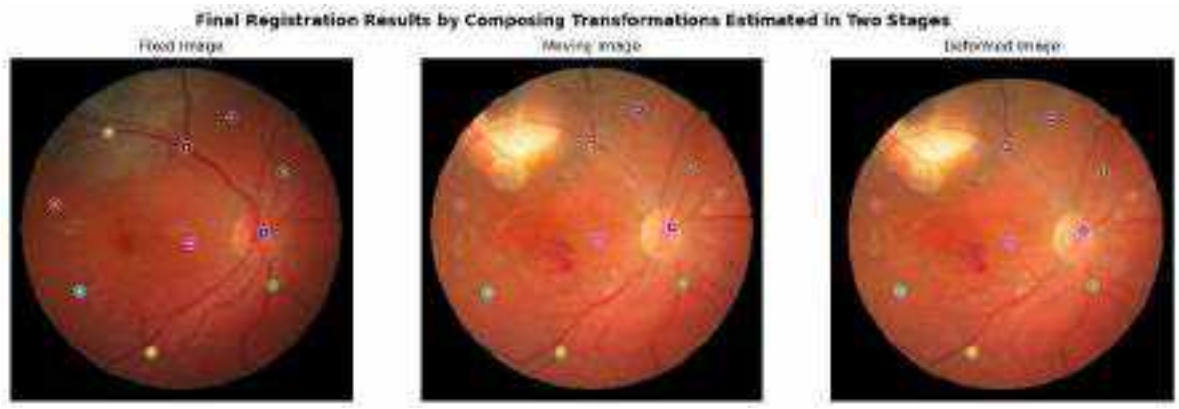



```
Homography Matrix:
[[ 9.63598783e-01 -3.75356216e-02  3.24943123e+01]
 [ 6.06765589e-03  9.30956119e-01  2.66330104e+01]
 [-2.36484260e-06 -3.93661175e-05  1.00000000e+00]]
```

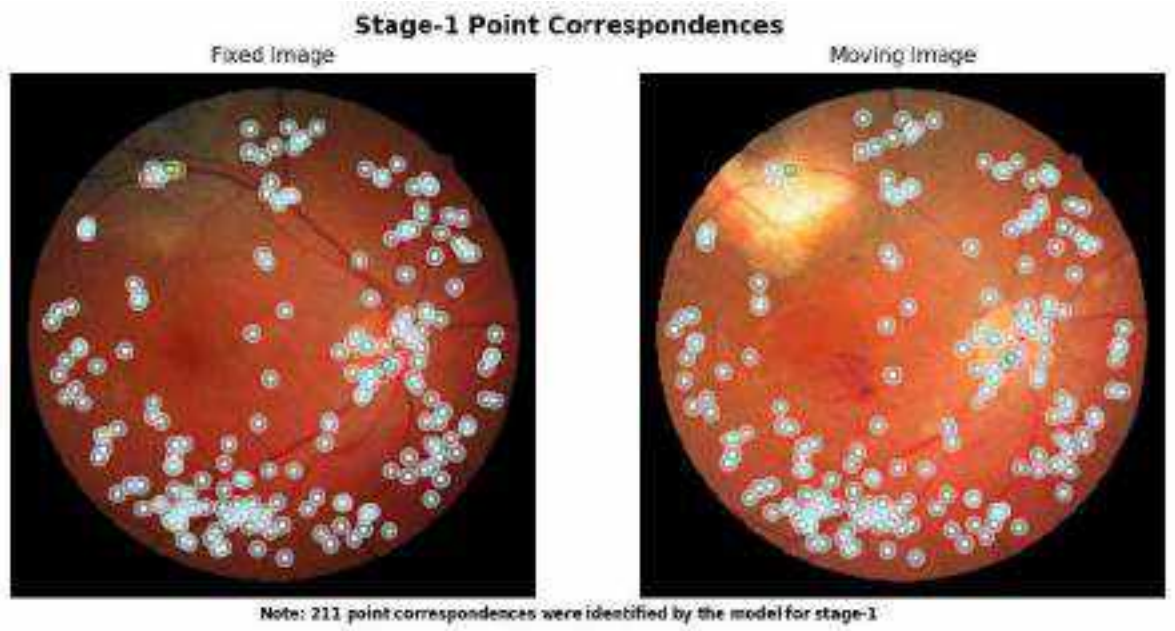



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



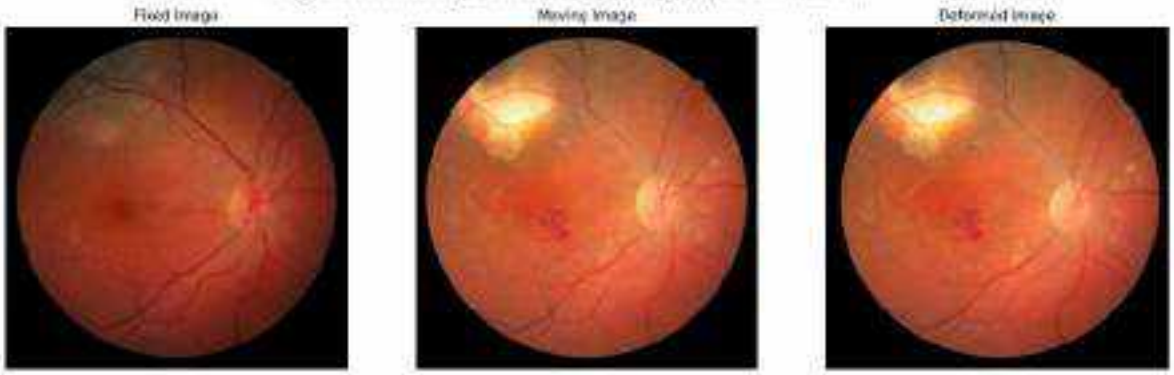



```
Mean Landmark Error for Case 7 Before Registration is 41.654178917603126 pixels
Mean Landmark Error for Case 7 After Registration is 9.649582399986247 pixels
Case 8
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A09_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A09_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

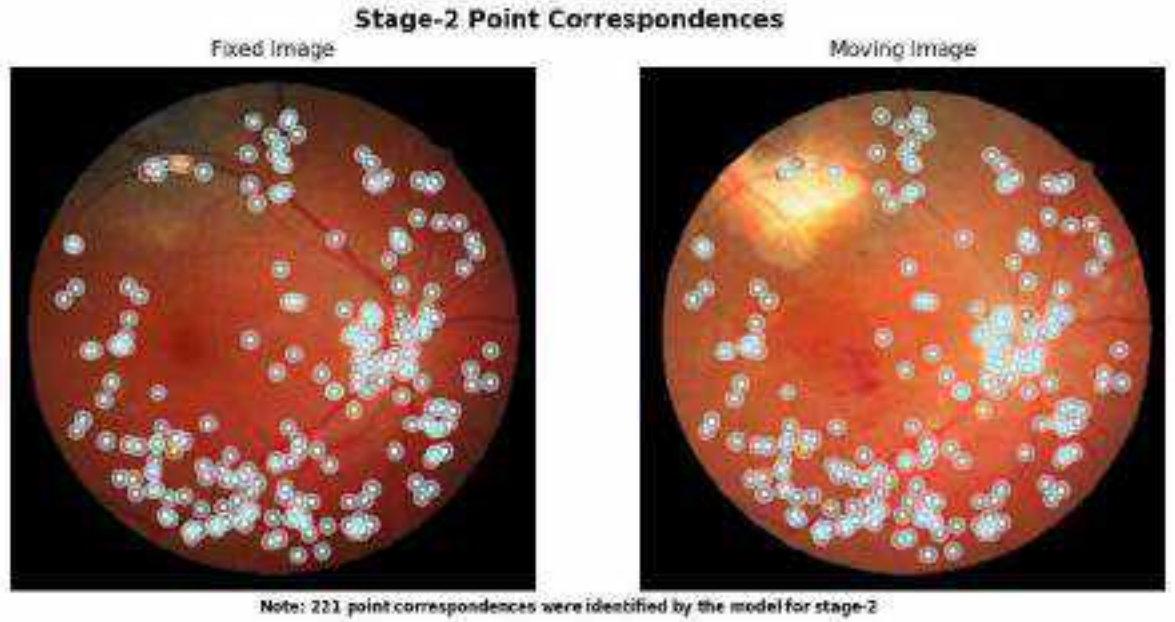


Note: 211 point correspondences were identified by the model for stage-1

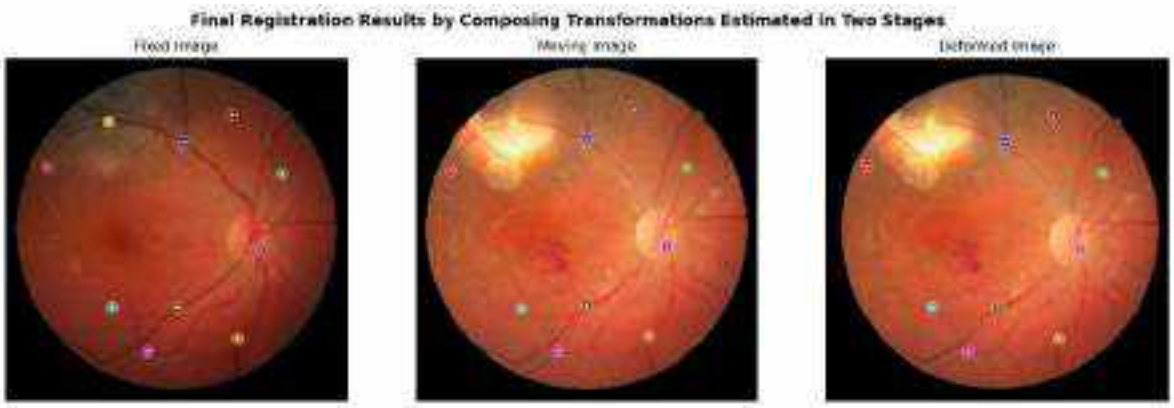
```
Homography Matrix:
[[ 1.00512191e+00 -4.36826719e-02  2.89012447e+01]
 [ 4.10401220e-02  9.82092862e-01 -6.34808245e+00]
 [ 1.50064203e-05 -1.22192350e-05  1.00000000e+00]]
```



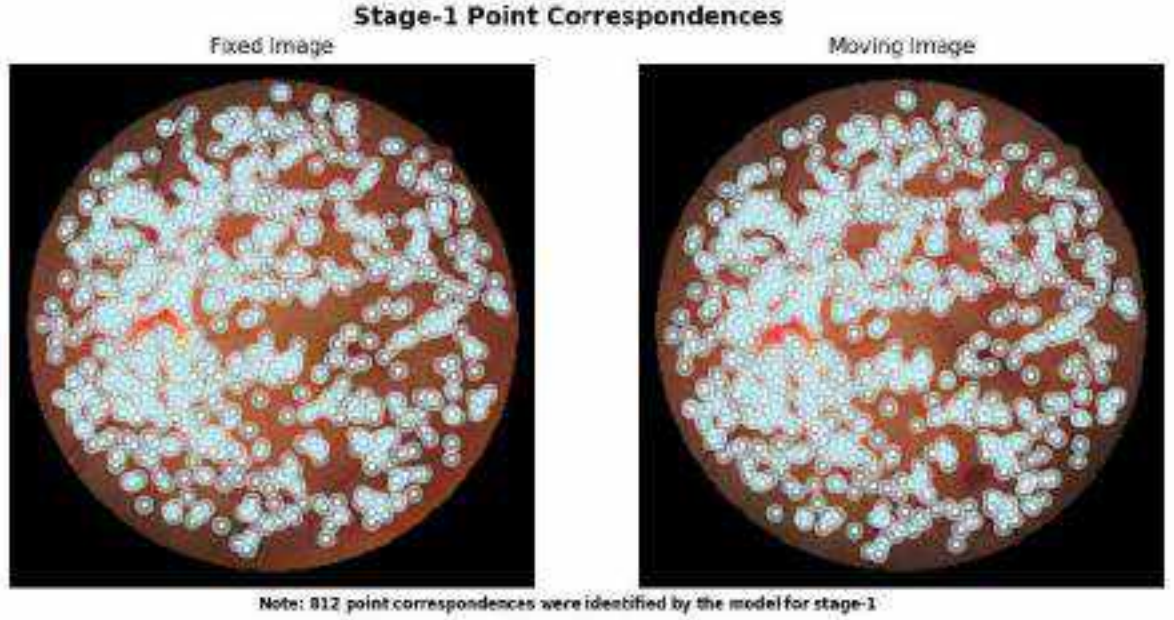

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



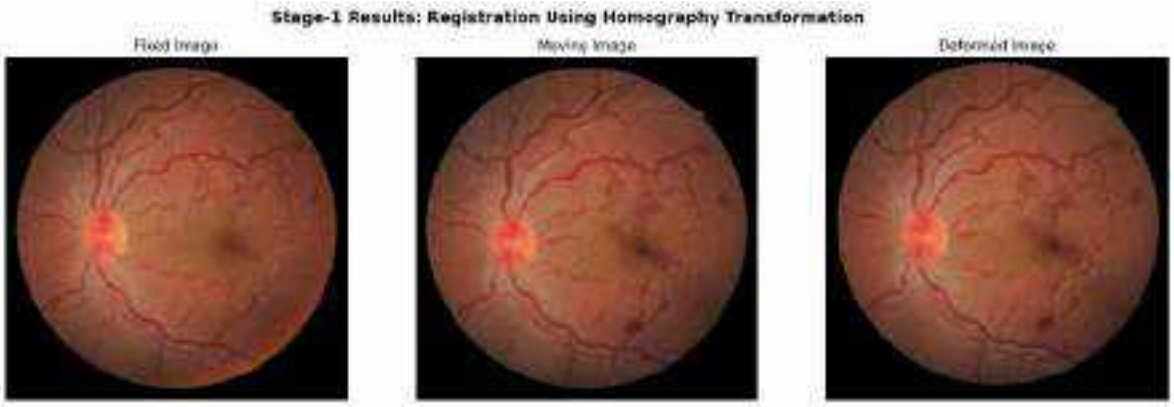
Note: 221 point correspondences were identified by the model for stage-2



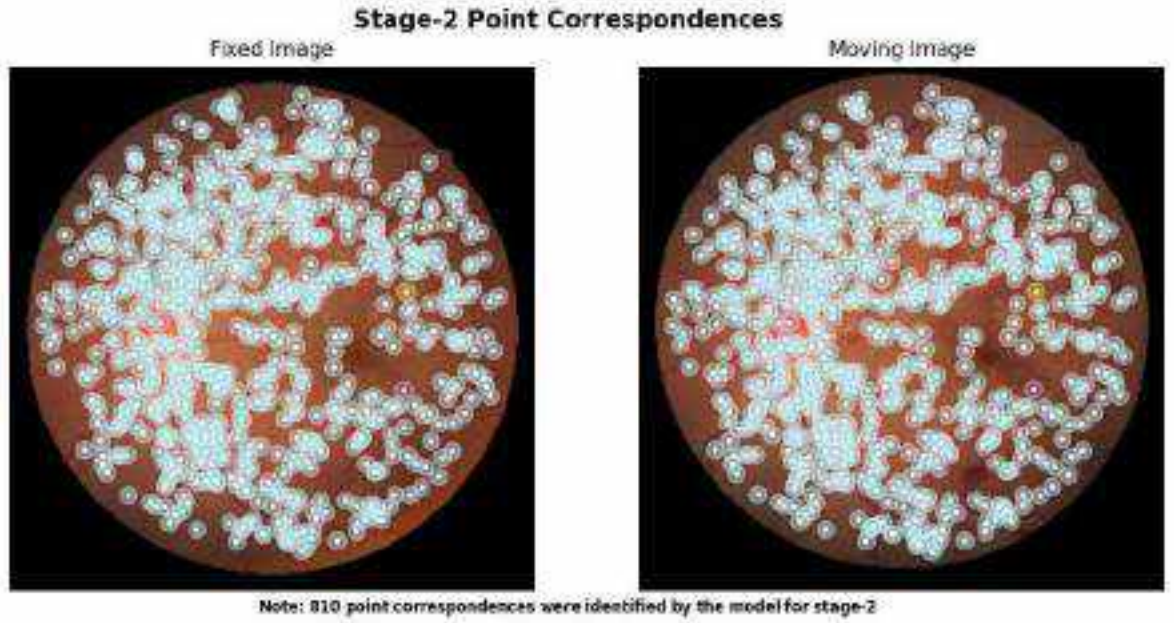

```
Mean Landmark Error for Case 8 Before Registration is 43.33436523051624 pixels
Mean Landmark Error for Case 8 After Registration is 4.894128419869896 pixels
Case 9
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A10_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A10_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



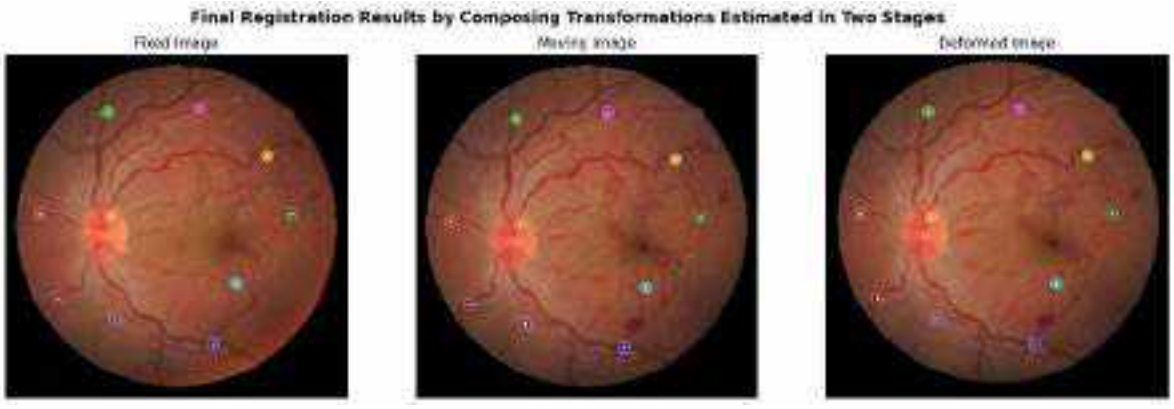

```
Homography Matrix:
[[ 1.00432208e+00 -1.64545100e-02  6.64612853e+00]
 [ 1.72838798e-02  1.00238069e+00 -2.35382779e+01]
 [ 1.14649035e-06 -6.01417052e-06  1.00000000e+00]]
```



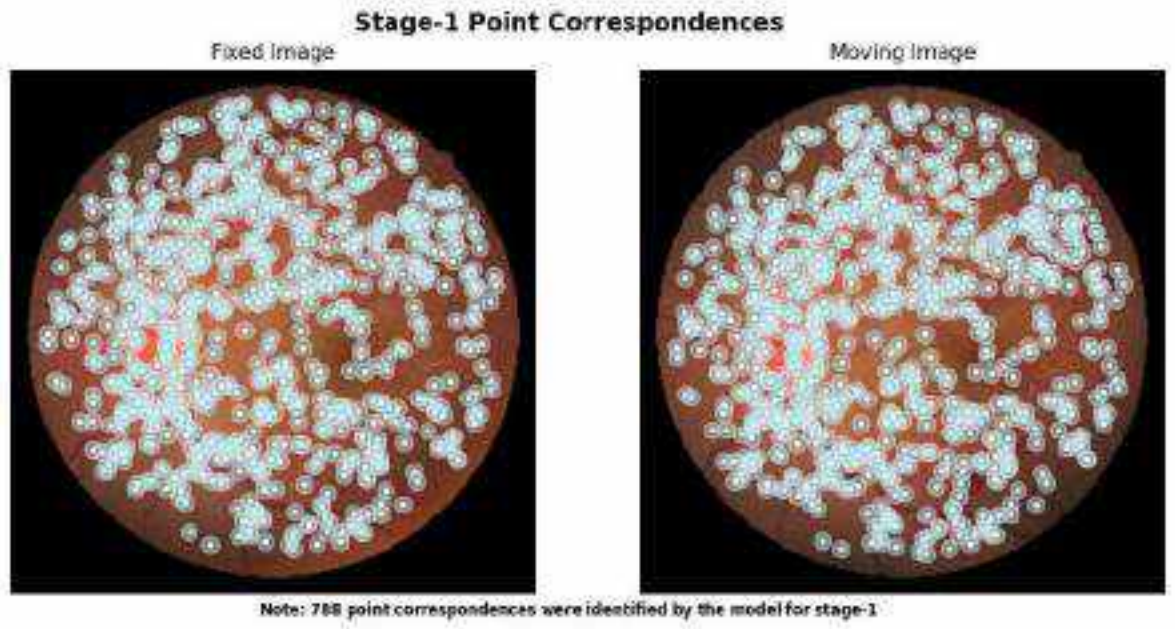

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



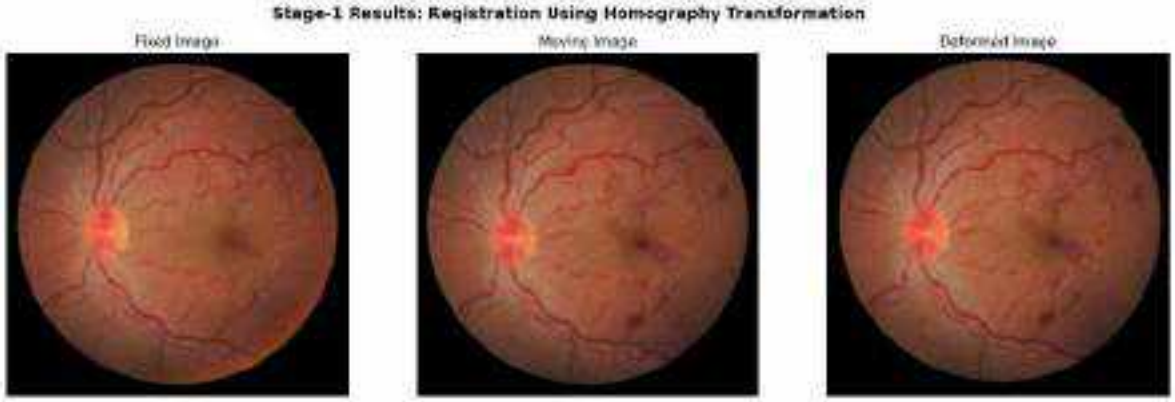
Note: 810 point correspondences were identified by the model for stage-2



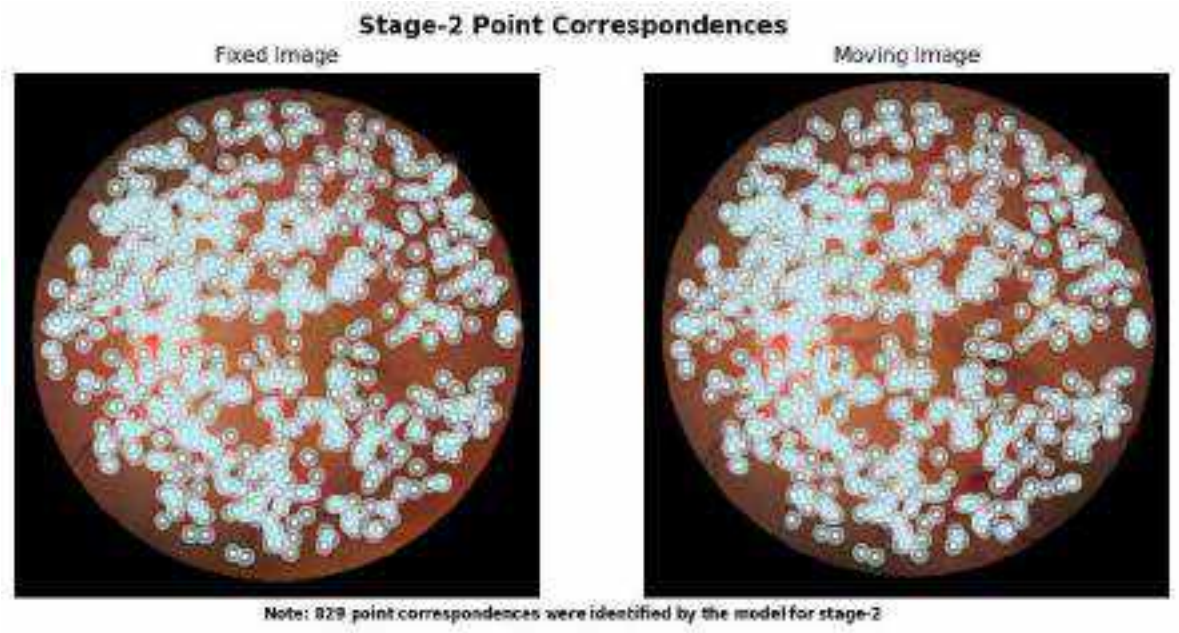
```
Mean Landmark Error for Case 9 Before Registration is 45.15990401104419 pixels
Mean Landmark Error for Case 9 After Registration is 3.0147436988193017 pixels
Case 10
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A11_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A11_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

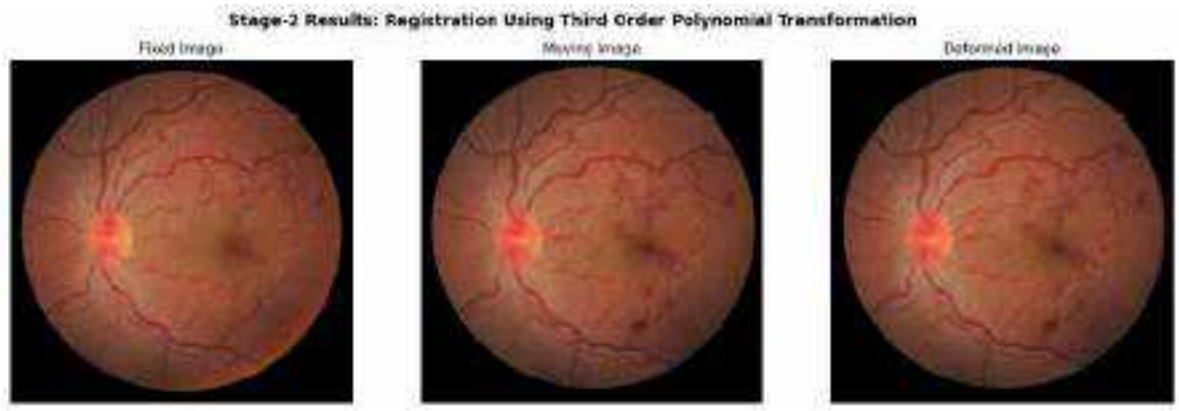


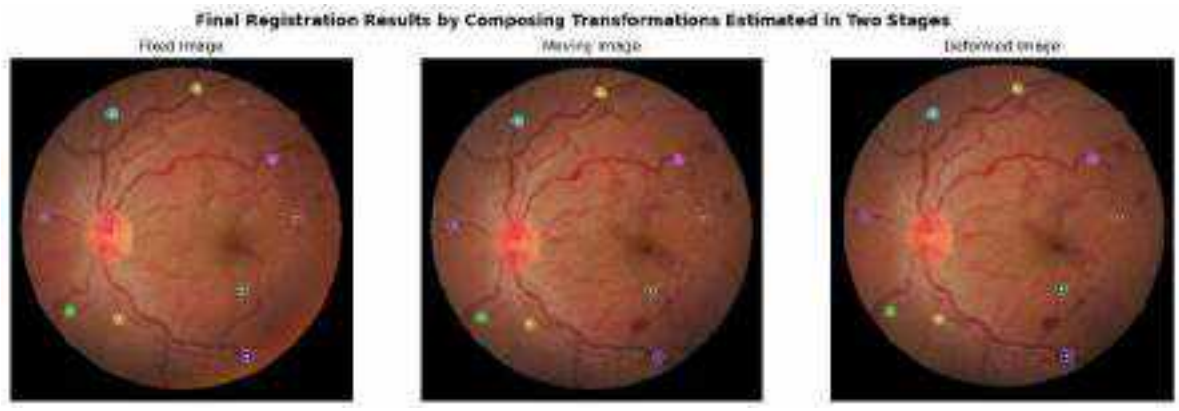

```
Homography Matrix:
[[ 1.01644656e+00 -3.13373059e-02  1.52969503e+01]
 [ 3.86867815e-02  1.00788401e+00 -3.04234414e+01]
 [ 1.31048505e-05 -4.76962083e-06  1.00000000e+00]]
```



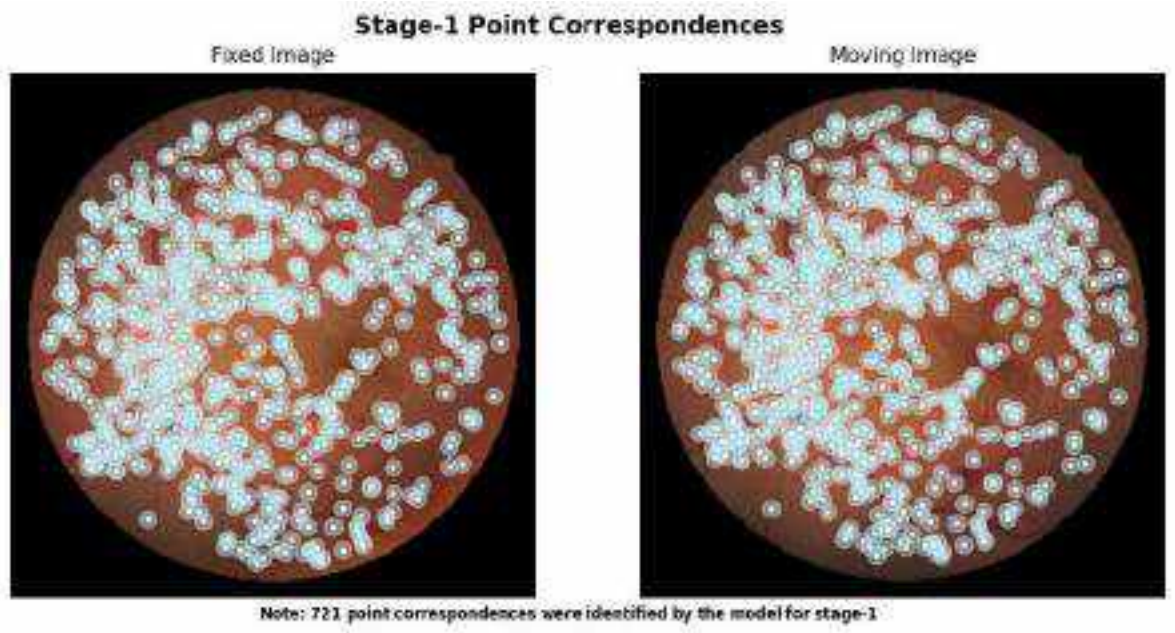

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```







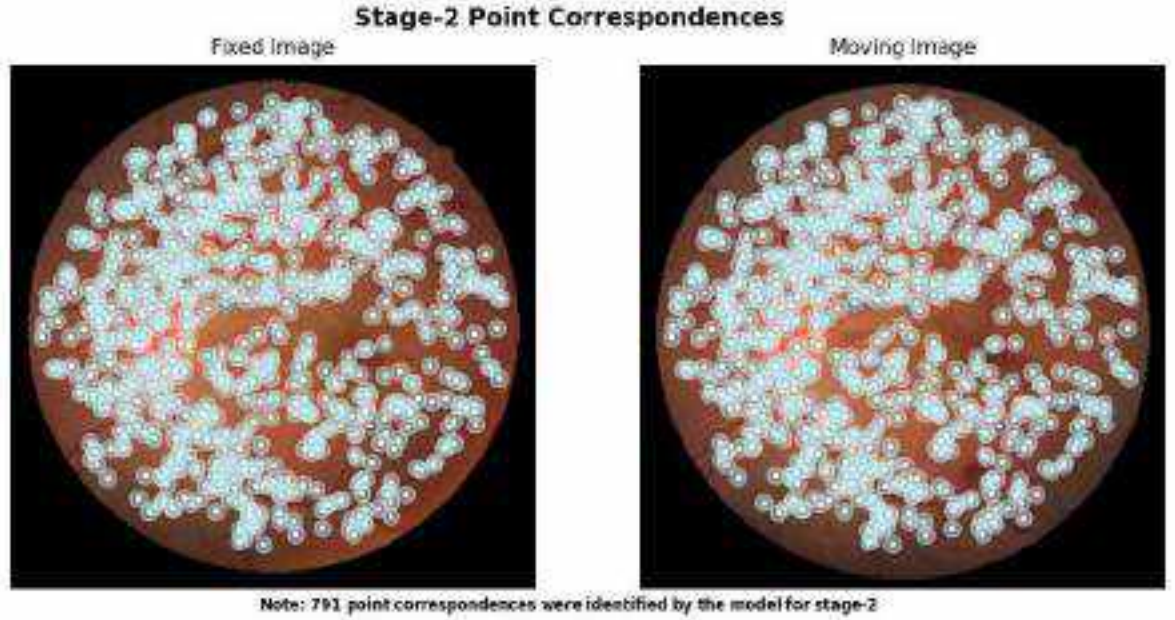

```
Mean Landmark Error for Case 10 Before Registration is 47.1435782371115 76 pixels
Mean Landmark Error for Case 10 After Registration is 2.6654641163835264 pixels
Case 11
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A12_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A12_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



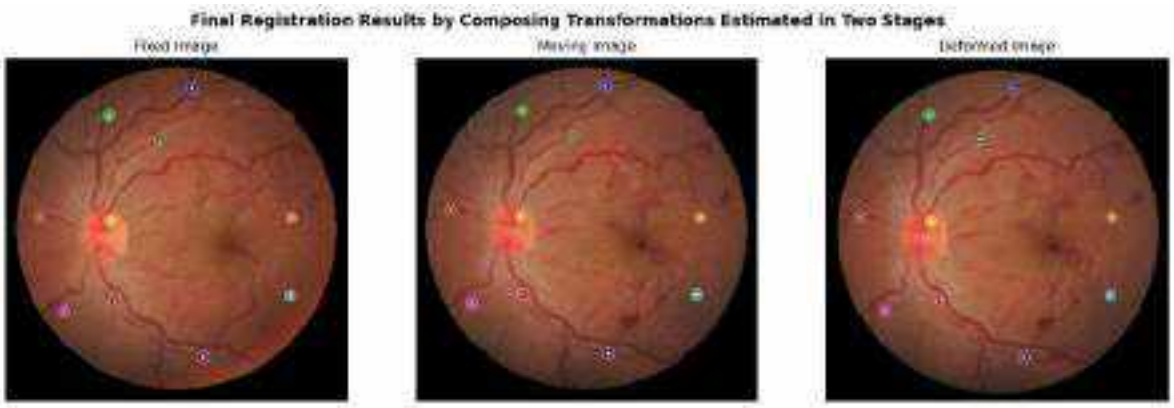
Note: 721 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.01502759e+00  3.13601001e-02 -1.55322597e+01]
 [-2.43019986e-02  1.01221081e+00  1.70267867e+01]
 [ 9.61688165e-06  6.99364820e-07  1.00000000e+00]]
```

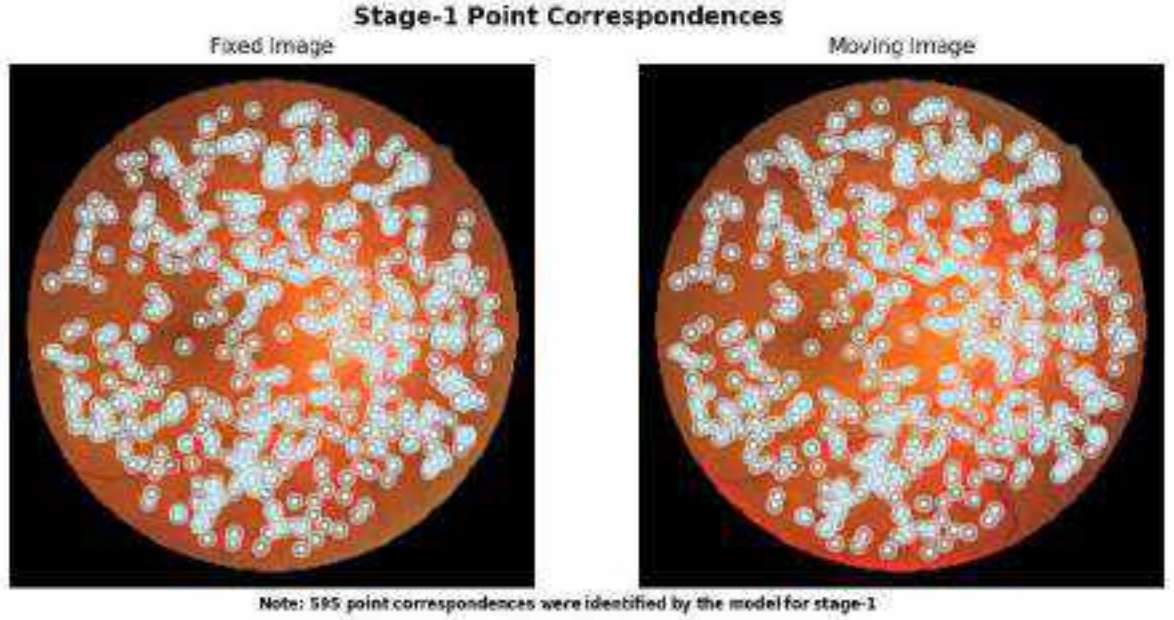

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



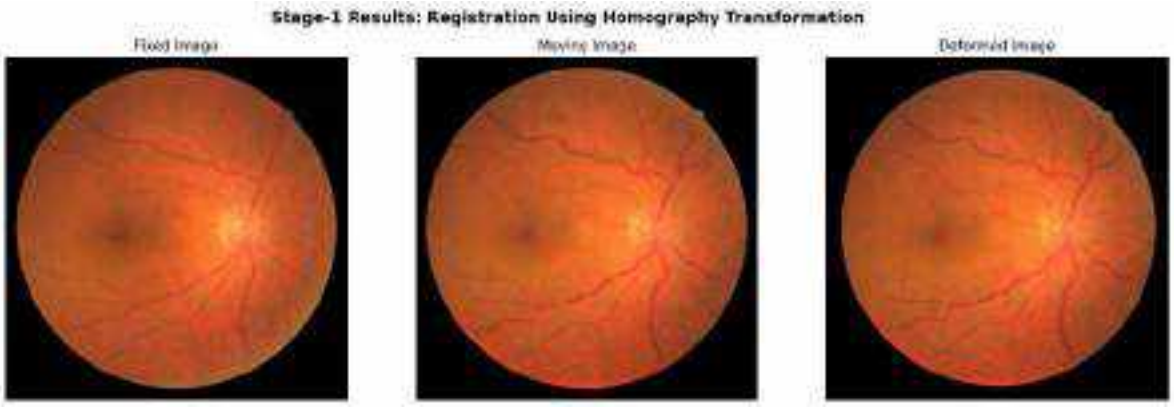
Note: 791 point correspondences were identified by the model for stage-2



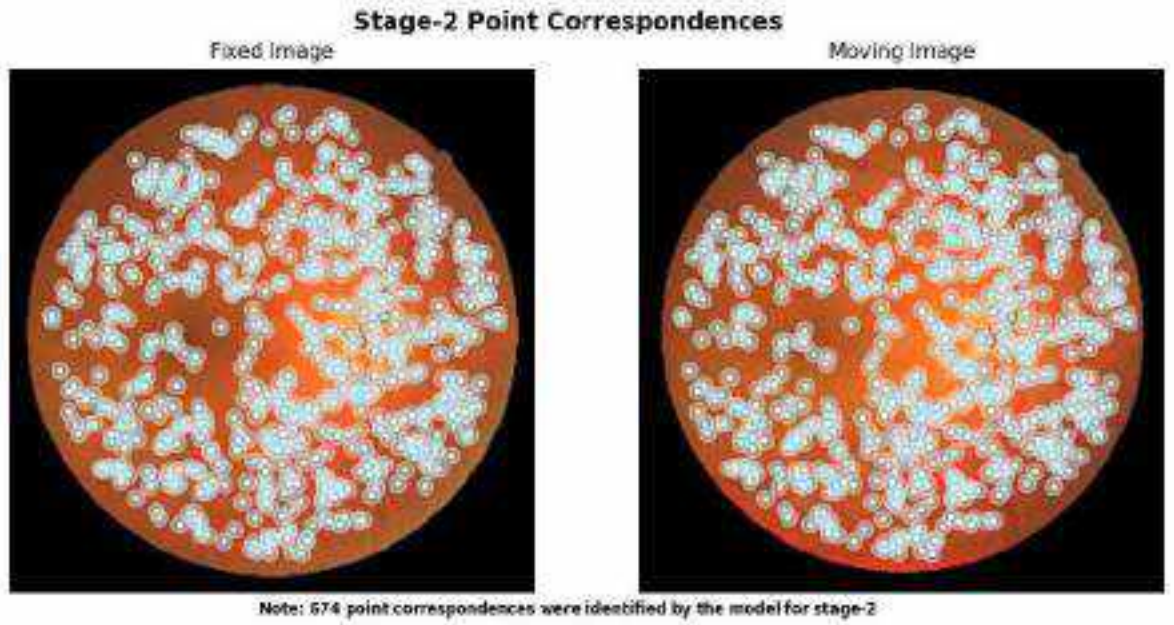

```
Mean Landmark Error for Case 11 Before Registration is 43.33058466932491 pixels
Mean Landmark Error for Case 11 After Registration is 2.940278346687 pixels
Case 12
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A13_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A13_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



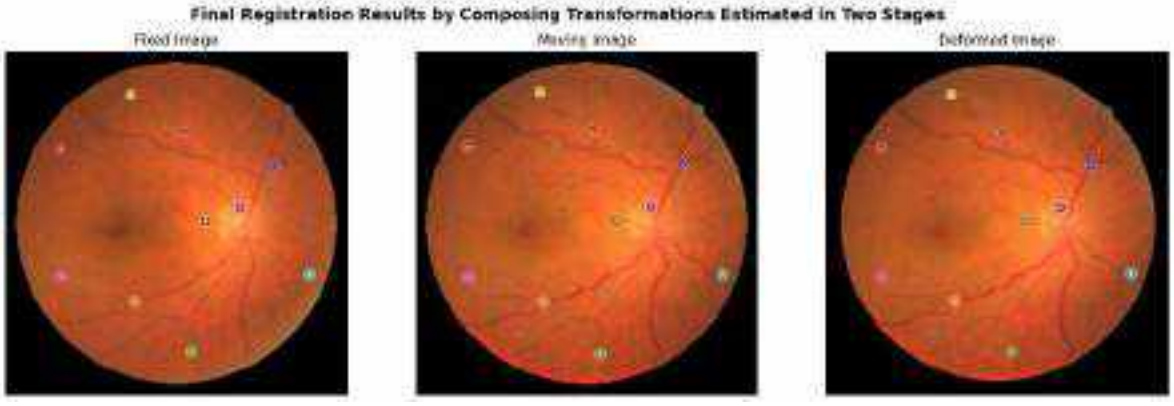

```
Homography Matrix:
[[ 9.81246949e-01  5.12279517e-03  7.87430672e+00]
 [-4.67705472e-03  9.86146415e-01  9.37333860e+00]
 [-5.89251950e-06  4.69220947e-06  1.00000000e+00]]
```



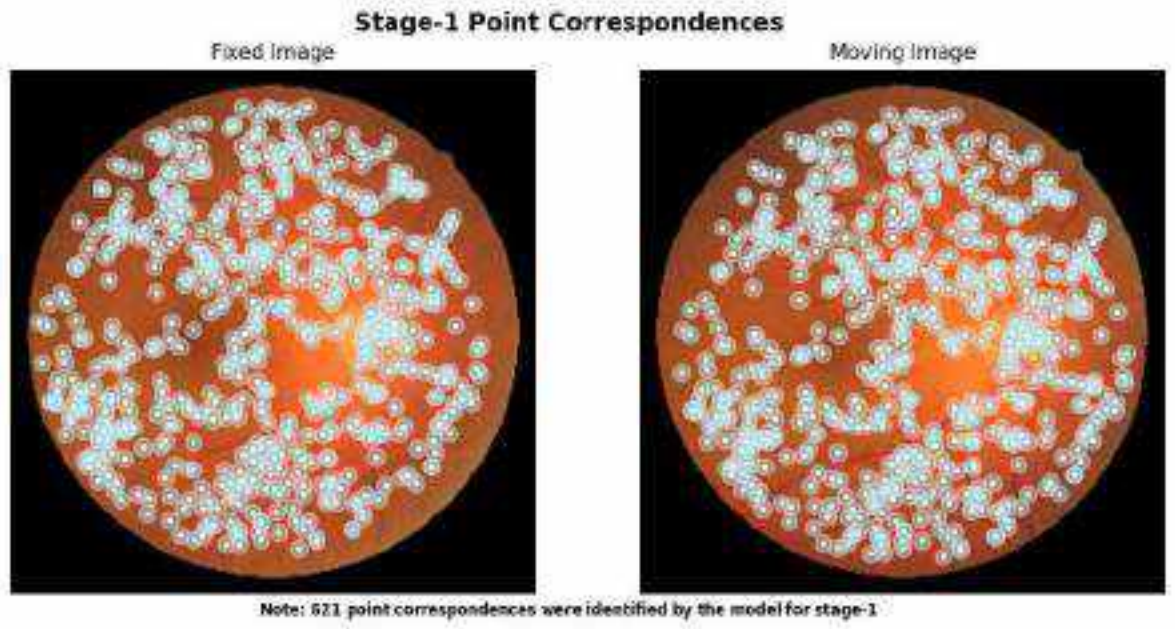

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 574 point correspondences were identified by the model for stage-2

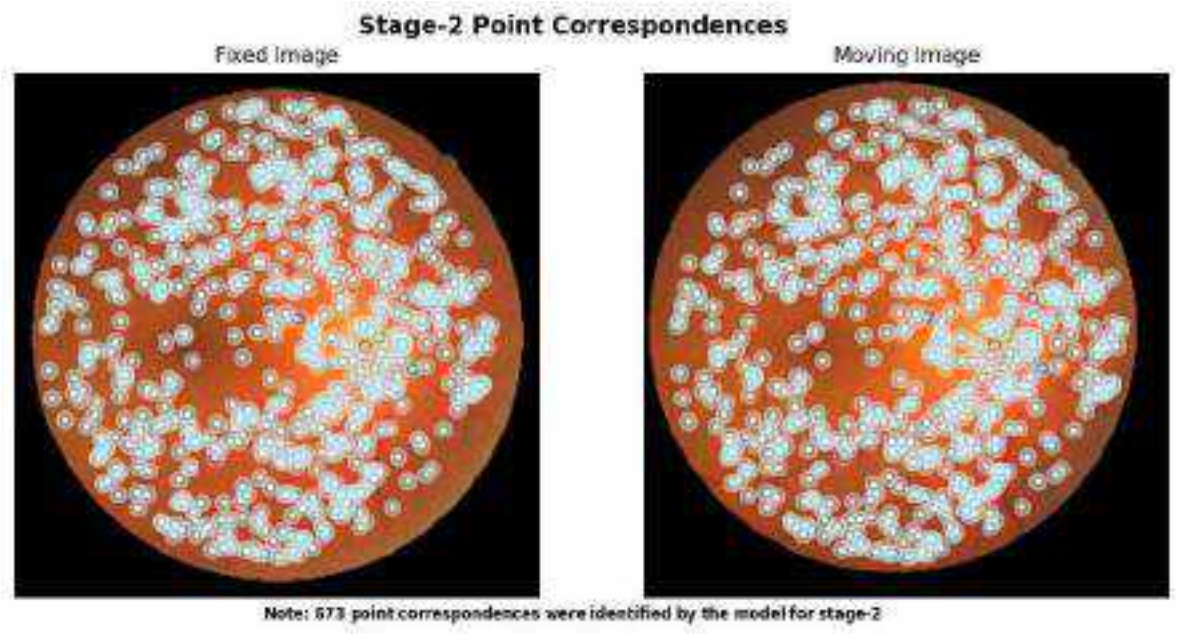


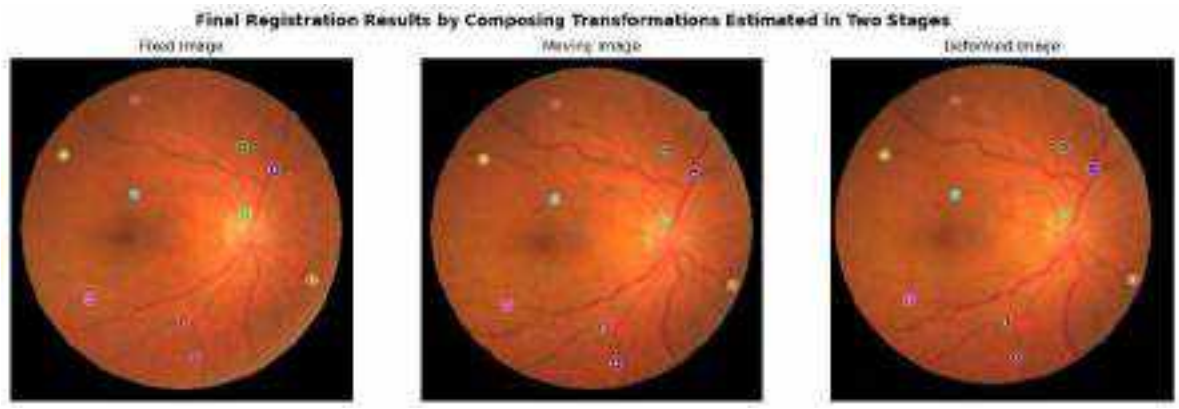
```
Mean Landmark Error for Case 12 Before Registration is 18.557587984837518 pixels
Mean Landmark Error for Case 12 After Registration is 4.5462253132750075 pixels
Case 13
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/A14_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/A14_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```




```
Homography Matrix:
[[ 9.76144212e-01  2.04889254e-03 -2.17011971e+01]
 [-8.91872570e-03  9.82056753e-01 -6.54251061e+00]
 [-1.50184774e-05 -7.76953720e-06  1.00000000e+00]]
```

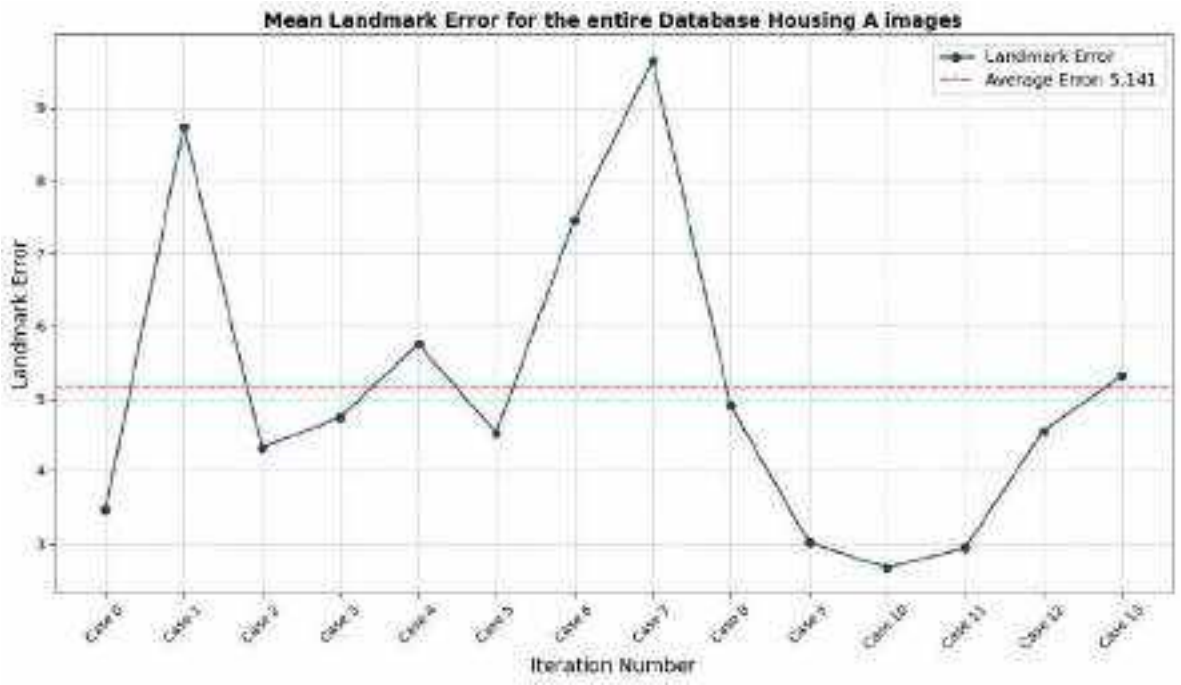

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



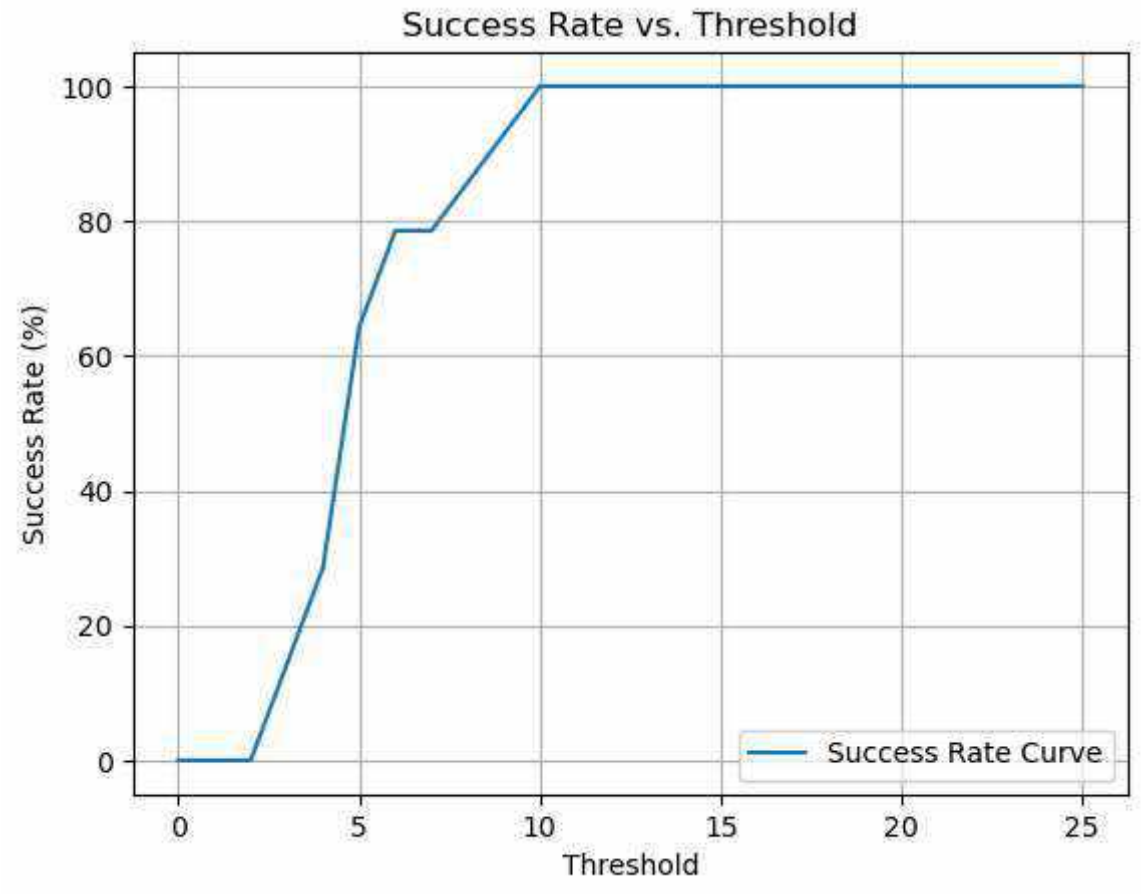



```
Mean Landmark Error for Case 13 Before Registration is 92.9675518560382 pixels
Mean Landmark Error for Case 13 After Registration is 5.3039634164823 25 pixels
```

In [20]:
```
plot_landmark_errors(landmark_errors1,os.path.join(os.getcwd(),'FIRE_Image_Regis
```



In [21]:
```
compute_plot_FIRE_AUC(landmark_errors1)
```



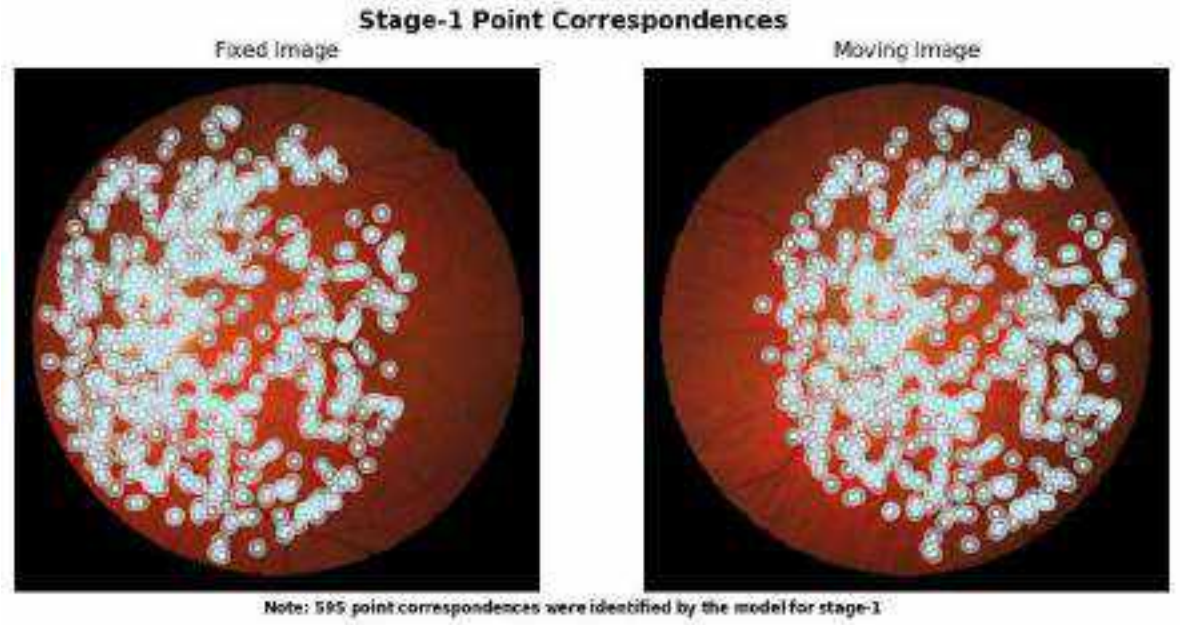

```
AUC: 0.8171428571428572
```

Class-P

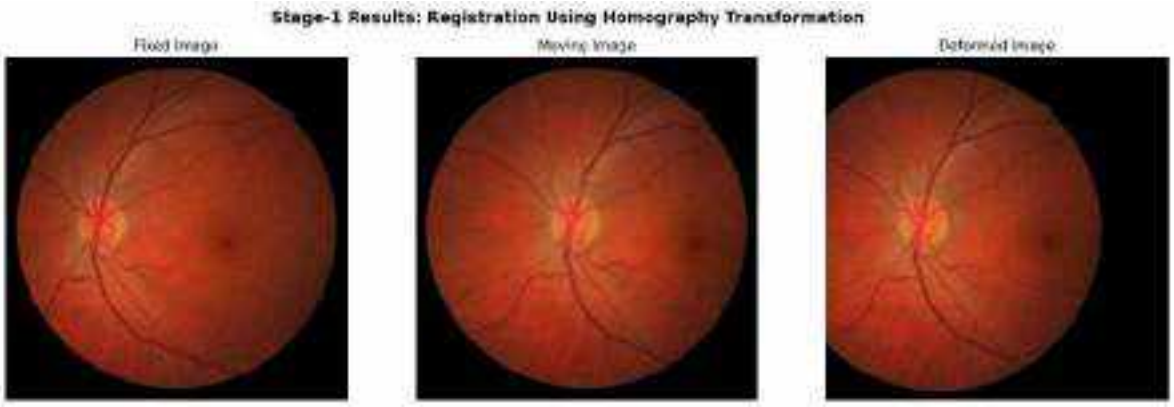
In [22]:
```
landmark_errors2=[]
for i in range(len(images_P)):
    print("Case {}".format(i))
    print("Loading Fixed Images {0} Moving Image{1} to the framework".format(ima
    original_low_res,computed_low_res = main(images_P[i],os.path.join(os.getcwd(
    imags,imgs,homography_matrix_low_res = compute_homography_matrix_and_plot(im
```

```
    if len(homography_matrix_low_res) !=0:
        transformed_points_hom = transform_points_homography(scaled_moving_point
        transformed_points_high_res_hom =  coordinates_rescaling(transformed_poi
        original_low_res,computed_low_res = main(imags,os.path.join(os.getcwd(),
        imag,imgs,polynomial_matrix_low_res = compute_third_order_polynomial_mat
        if len(polynomial_matrix_low_res) !=0:
            ## rescaled version for dispaly purposes
            transformed_points_poly = transform_points_third_order_polynomial(tr
            original_image_point_correspondences(imag,images_P[i][0],img_size, s
            ### Original Version for computation of errors
            polynomial_matrix = transform_points_third_order_polynomial_matrix(c
            bef_error = compute_landmark_error(fixed_points_P[i],fixed_image_siz
            aft_error = compute_landmark_error_fixed_space(polynomial_matrix,fix
            print("Mean Landmark Error for Case {0} Before Registration is {1} p
            print("Mean Landmark Error for Case {0} After Registration is {1} pi
            landmark_errors2.append(aft_error)
        else:
            landmark_errors2.append(10000)
    else:
        landmark_errors2.append(10000)
```

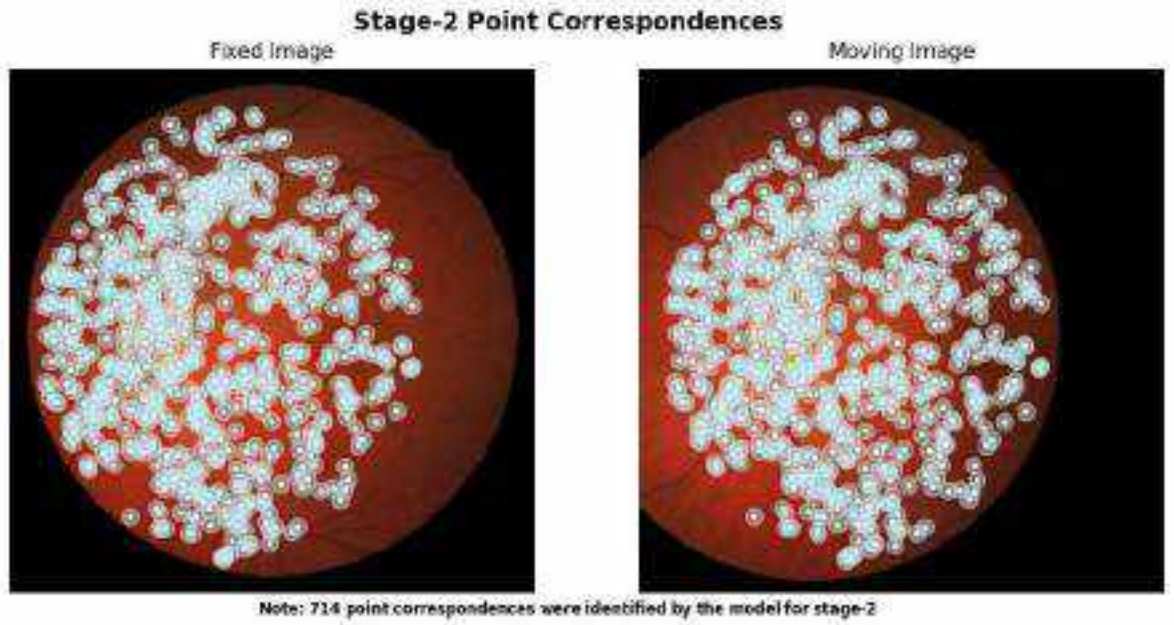
```
Case 0
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P01_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P01_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
/scratch/local/29752099/ipykernel_2239179/635223464.py:100: UserWarning: To copy
construct from a tensor, it is recommended to use sourceTensor.clone().detach() o
r sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(so
urceTensor).
  points_indices = torch.tensor(pts_list)
```



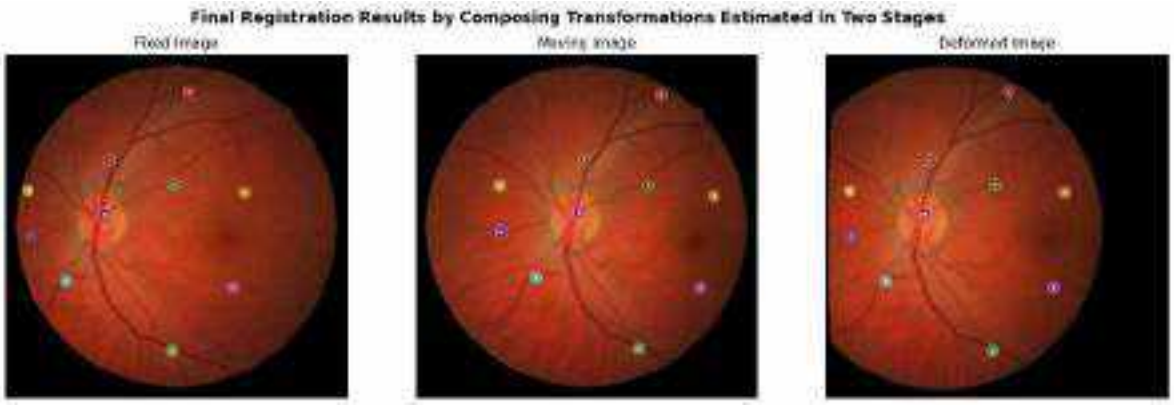

```
Homography Matrix:
[[ 9.36501620e-01  2.93577504e-02 -1.63770253e+02]
 [-7.00610041e-02  9.61099369e-01  3.62162063e+01]
 [-8.04209275e-05 -6.60271396e-06  1.00000000e+00]]
```



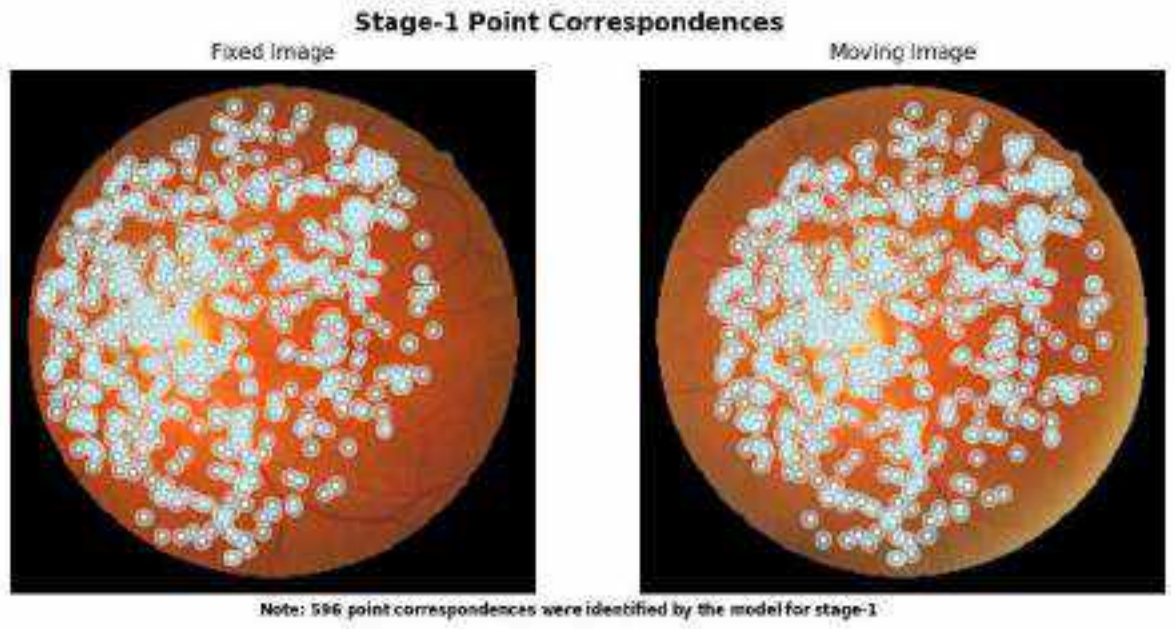

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



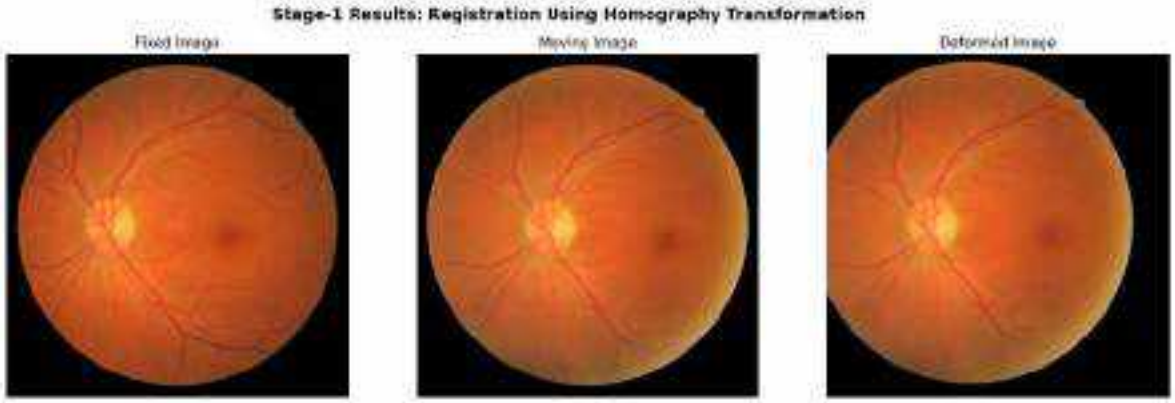
Note: 714 point correspondences were identified by the model for stage-2



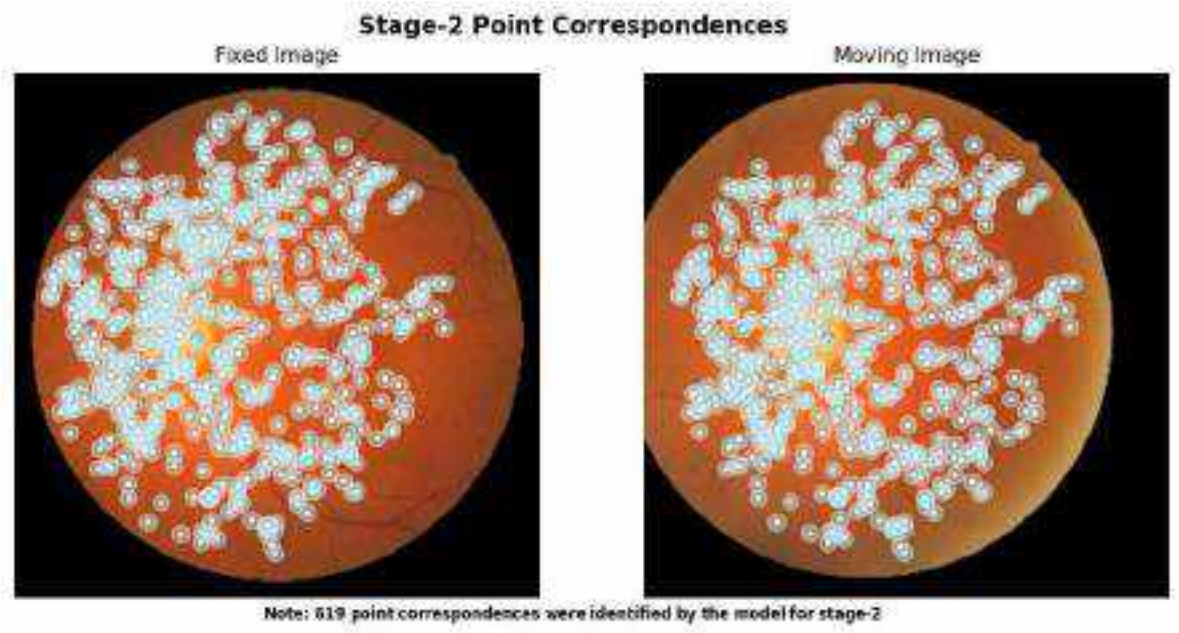
```
Mean Landmark Error for Case 0 Before Registration is 518.9707408449206 pixels
Mean Landmark Error for Case 0 After Registration is 3.1673459368934935 pixels
Case 1
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P02_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P02_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

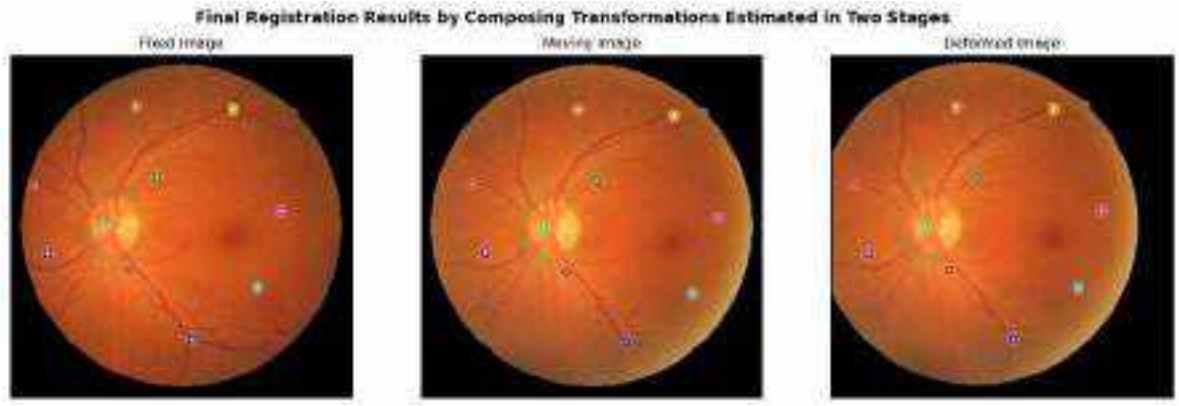



```
Homography Matrix:
[[ 9.76142991e-01  2.84156975e-02 -8.52023433e+01]
 [-4.57301940e-02  9.88677472e-01  1.14222576e+01]
 [-3.16354369e-05 -2.27021036e-06  1.00000000e+00]]
```

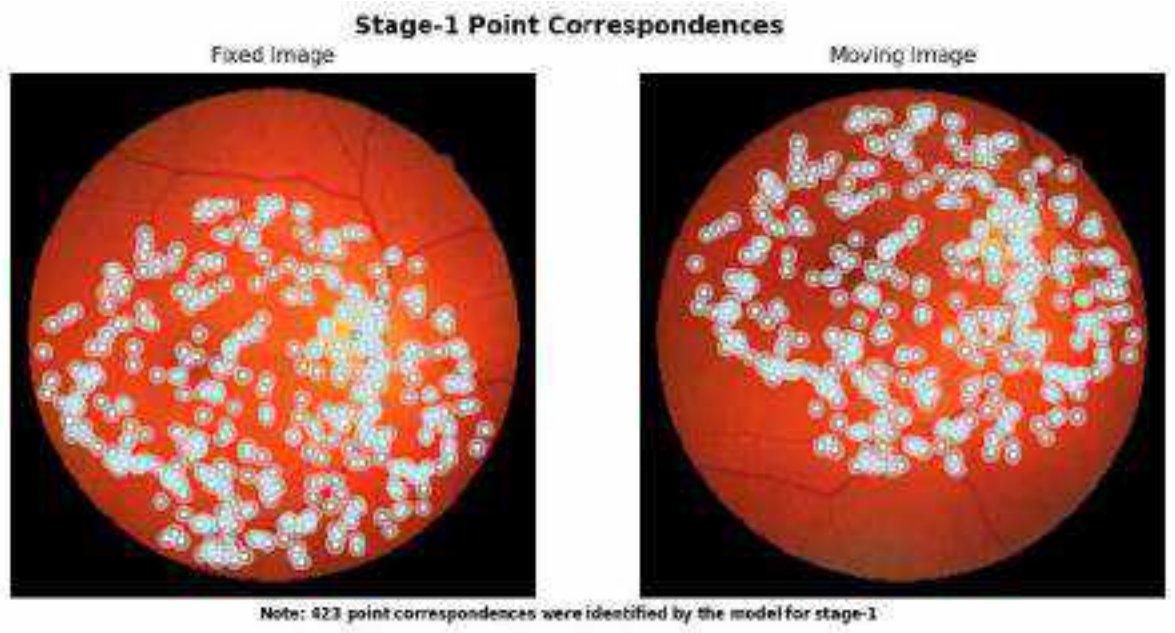



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





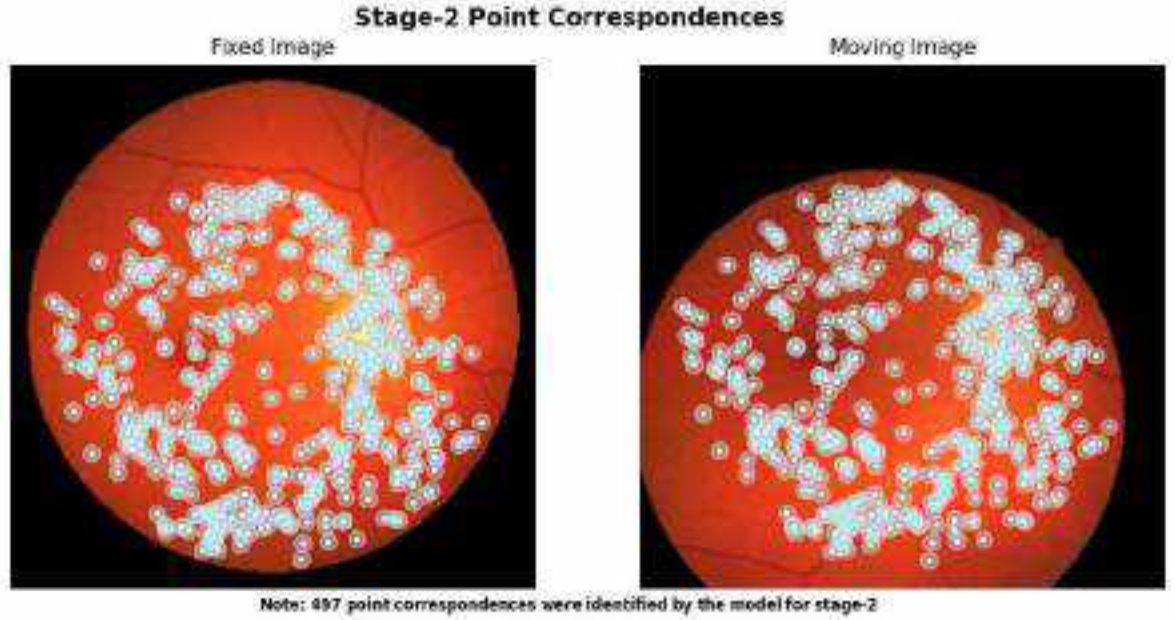

```
Mean Landmark Error for Case 1 Before Registration is 242.70910478555075 pixels
Mean Landmark Error for Case 1 After Registration is 2.1495385988038698 pixels
Case 2
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P03_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P03_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



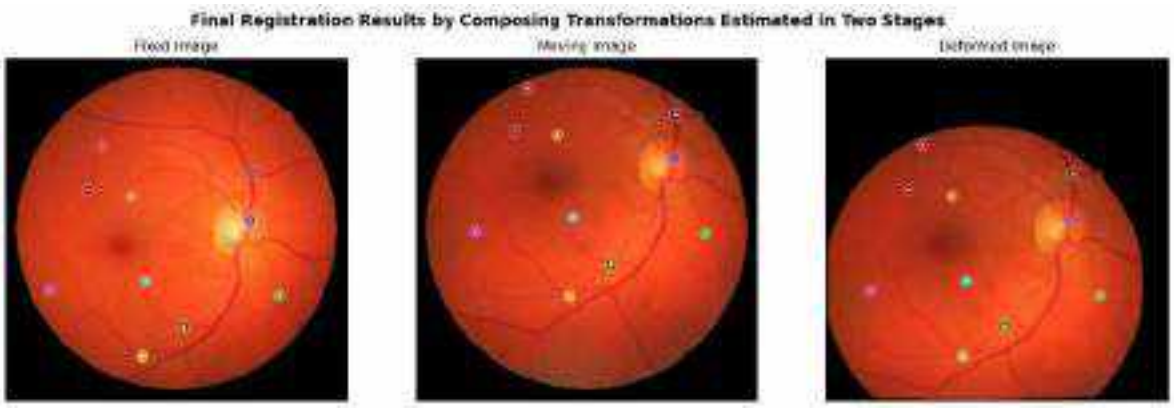
Note: 423 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.01434534e+00  2.38335494e-02 -5.17190437e+01]
 [-4.70274443e-03  1.00627559e+00  1.53952787e+02]
 [-2.40144179e-05  7.96783015e-05  1.00000000e+00]]
```

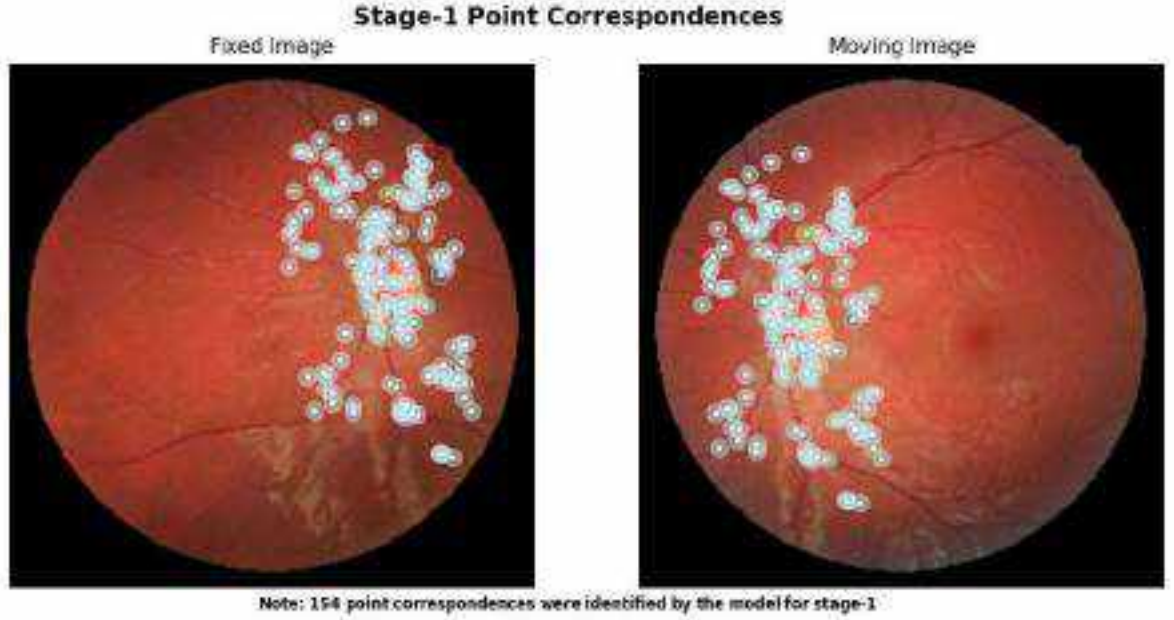

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



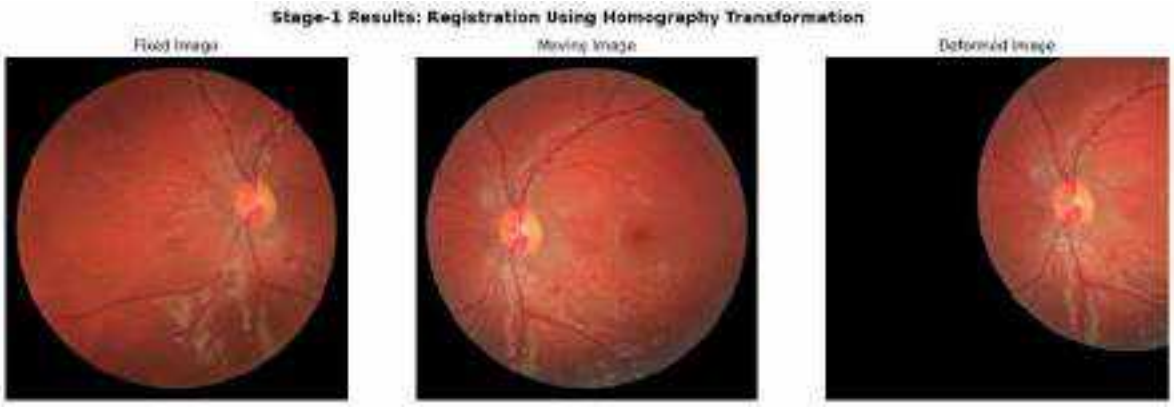
Note: 497 point correspondences were identified by the model for stage-2



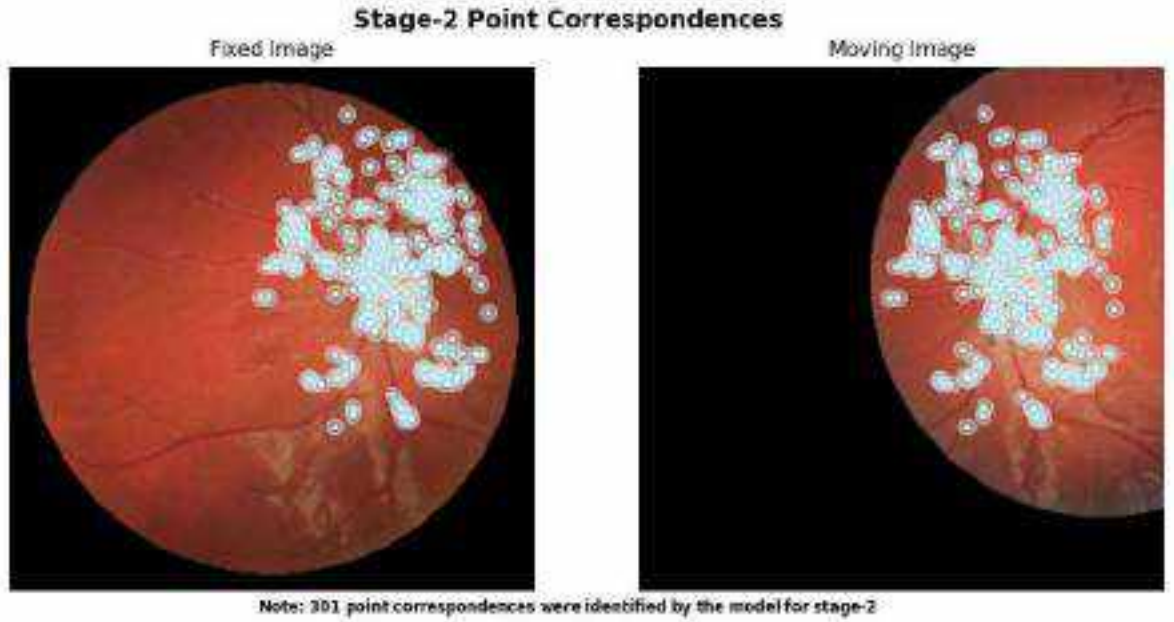

```
Mean Landmark Error for Case 2 Before Registration is 533.6799347117249 pixels
Mean Landmark Error for Case 2 After Registration is 3.200296161359284 pixels
Case 3
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P04_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P04_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



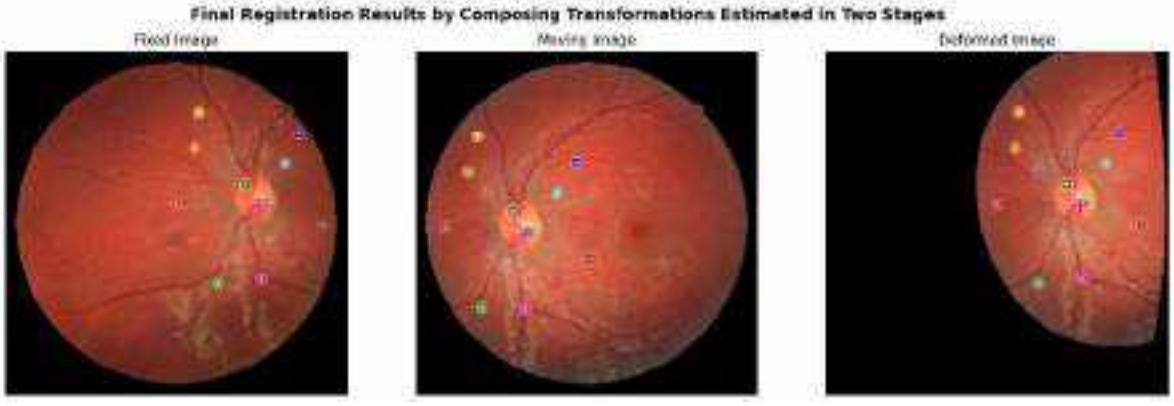

```
Homography Matrix:
[[ 1.13522559e+00  7.17142990e-02  3.39169927e+02]
 [-2.01496480e-03  1.01935689e+00 -6.57029392e+01]
 [ 1.78308787e-04 -1.51637873e-05  1.00000000e+00]]
```



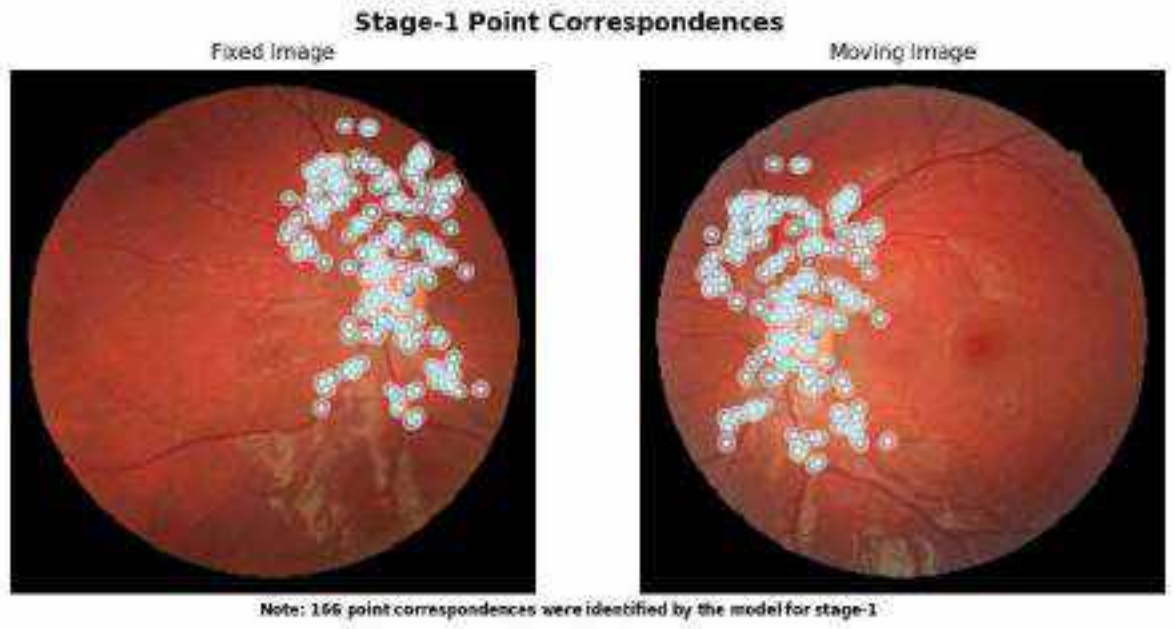

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



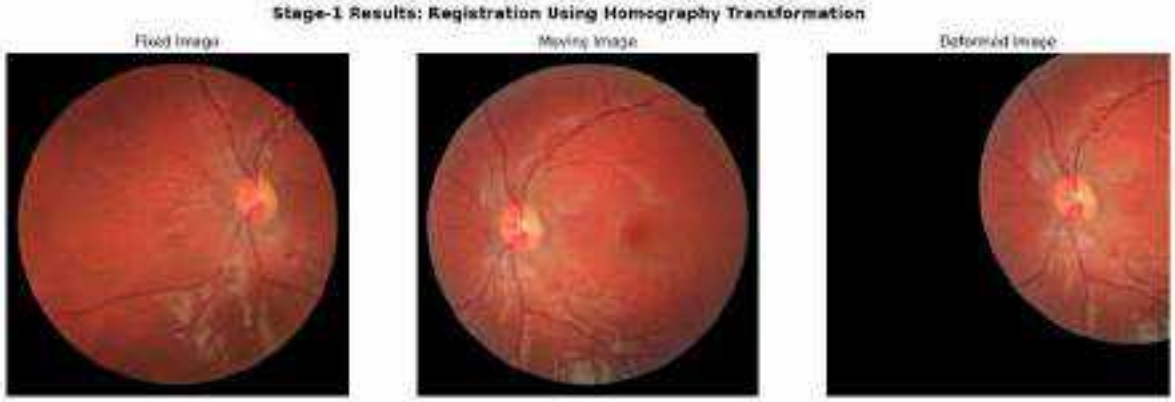
Note: 301 point correspondences were identified by the model for stage-2



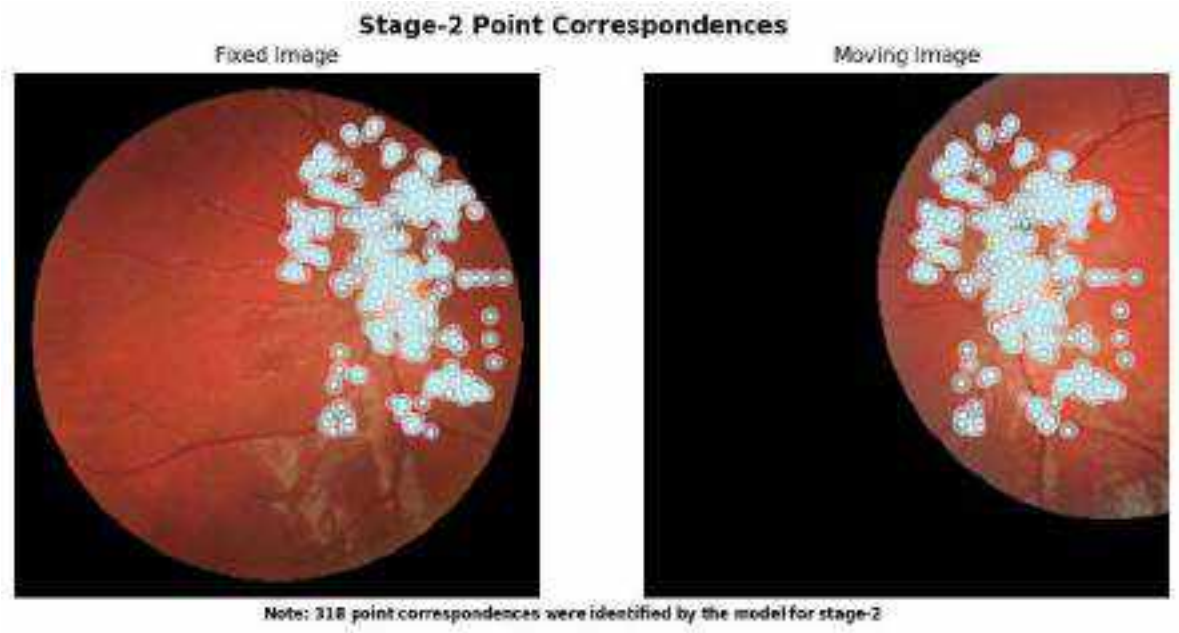
```
Mean Landmark Error for Case 3 Before Registration is 1216.778667252399 pixels
Mean Landmark Error for Case 3 After Registration is 4.886182542804362 pixels
Case 4
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P05_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P05_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

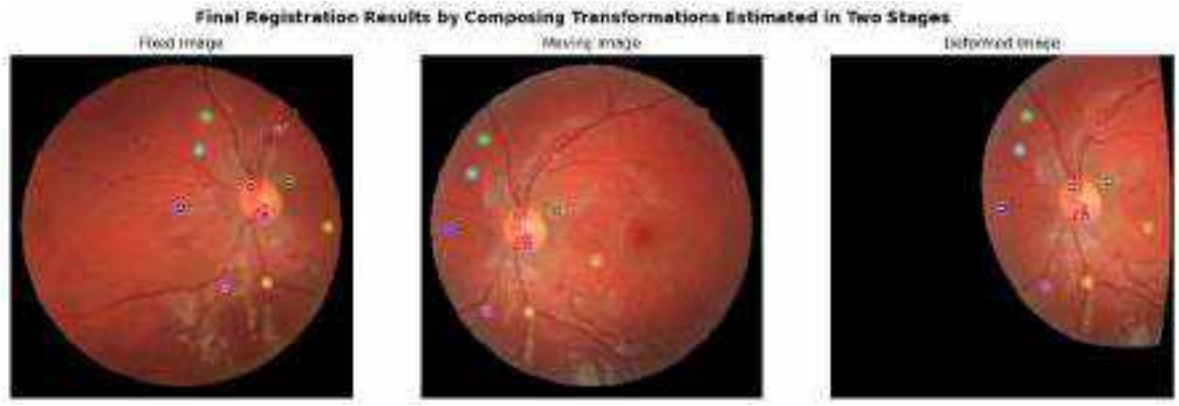



```
Homography Matrix:
[[ 1.14597085e+00  1.05786267e-01  3.31205059e+02]
 [-8.65920926e-03  1.02815164e+00 -6.53717739e+01]
 [ 1.85701608e-04  1.25671846e-06  1.00000000e+00]]
```

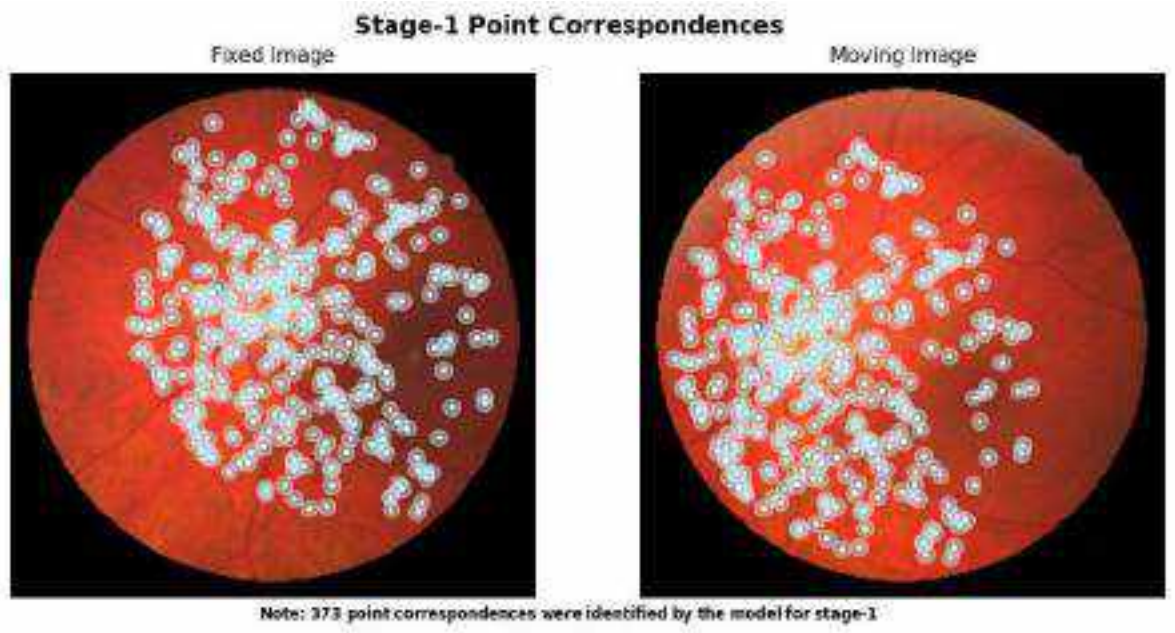



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



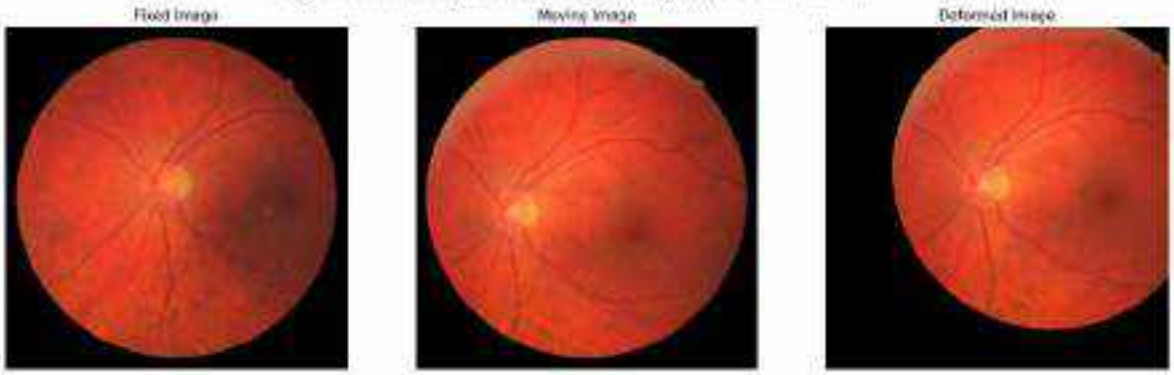



```
Mean Landmark Error for Case 4 Before Registration is 1228.2384174239191 pixels
Mean Landmark Error for Case 4 After Registration is 5.217950574721656 pixels
Case 5
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P06_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P06_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

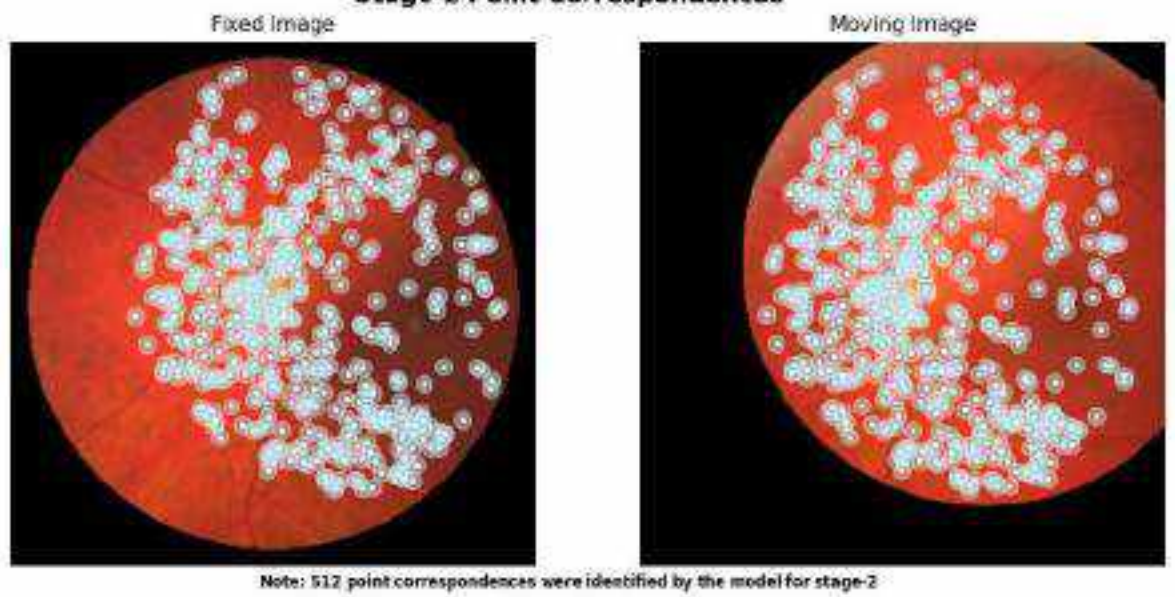


Note: 373 point correspondences were identified by the model for stage-1

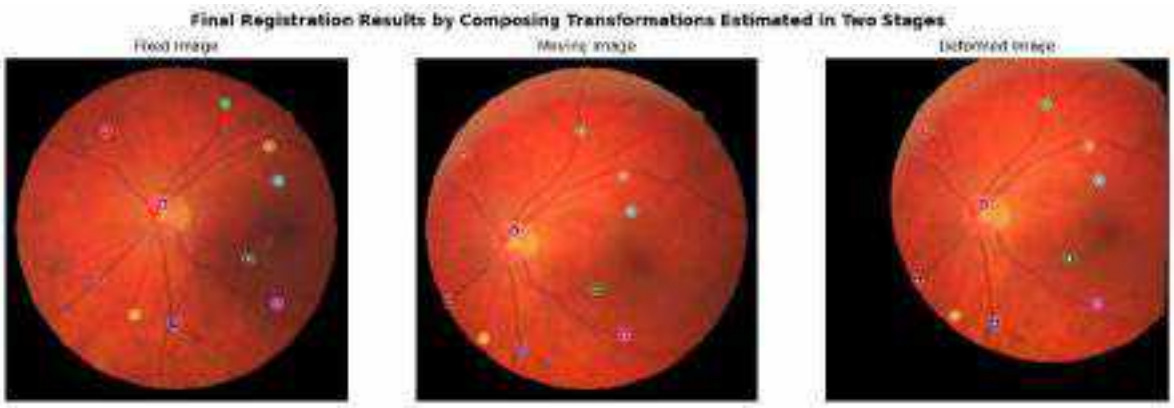
```
Homography Matrix:
[[ 1.06415204e+00  2.08883661e-02  1.34354978e+02]
 [-4.82486148e-03  9.91867980e-01 -6.55668276e+01]
 [ 9.37027990e-05 -4.50294785e-05  1.00000000e+00]]
```



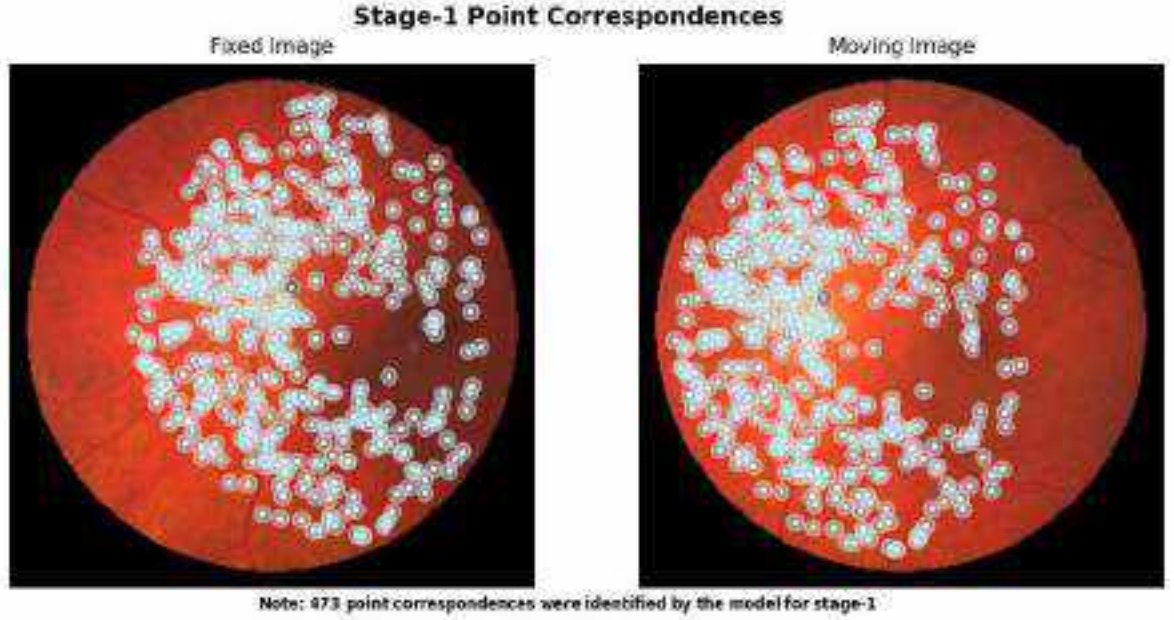

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



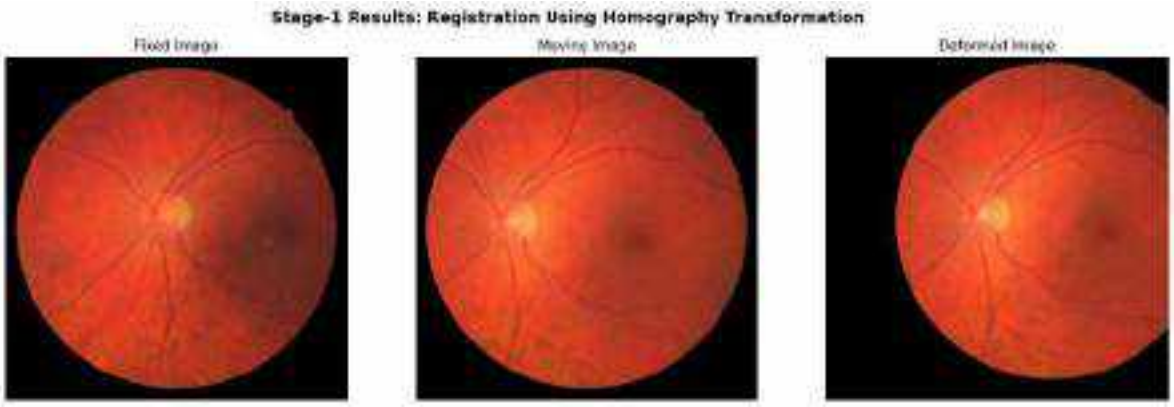
Note: 512 point correspondences were identified by the model for stage-2



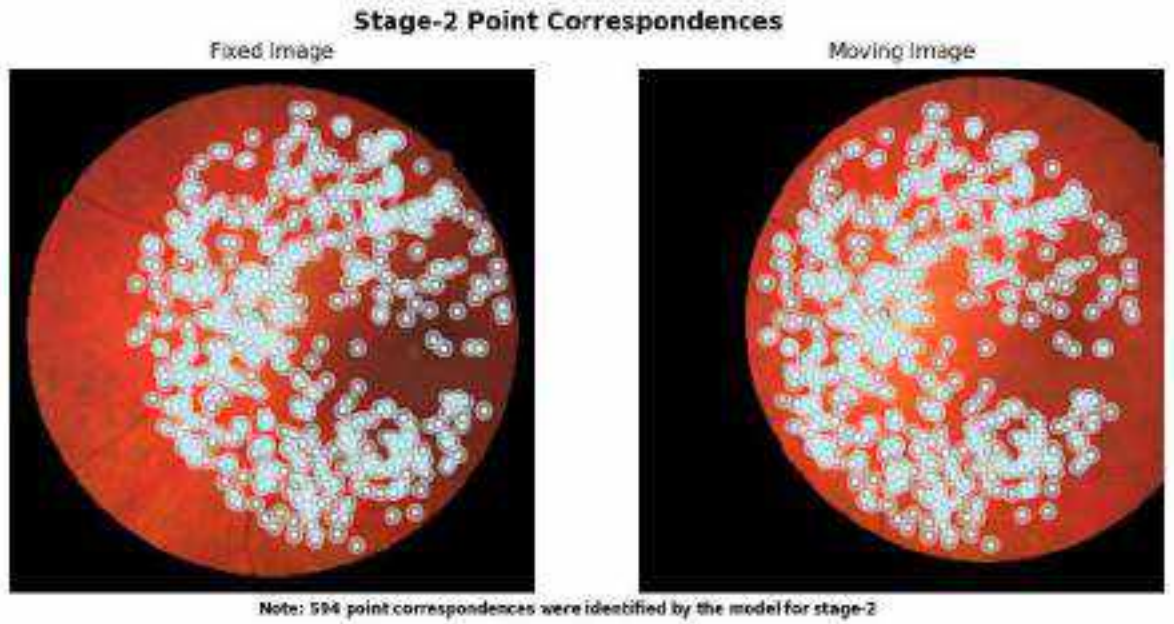

```
Mean Landmark Error for Case 5 Before Registration is 553.3271765541044 pixels
Mean Landmark Error for Case 5 After Registration is 4.21411116858552 pixels
Case 6
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P07_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P07_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



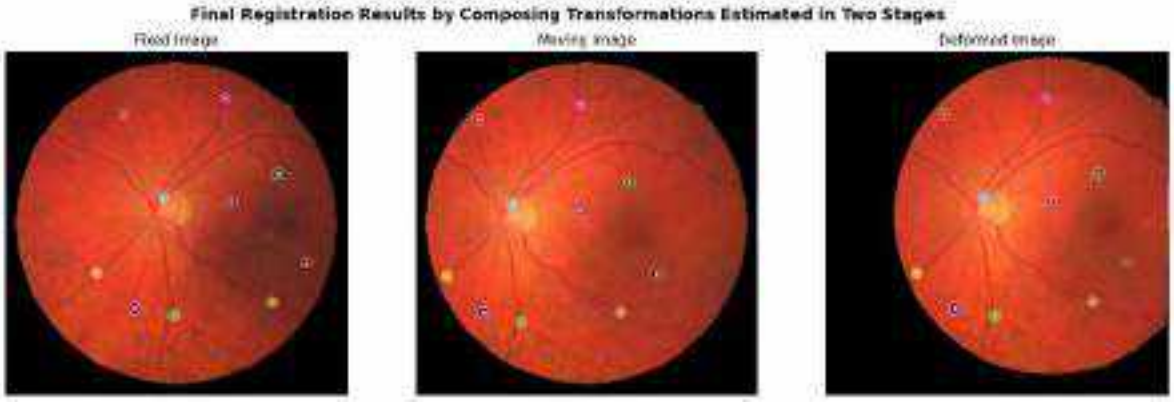

```
Homography Matrix:
[[ 1.06852175e+00  2.50124574e-02  1.43085285e+02]
 [ 1.31608995e-02  1.01651612e+00 -1.93299751e+01]
 [ 9.39442589e-05 -1.29593654e-05  1.00000000e+00]]
```



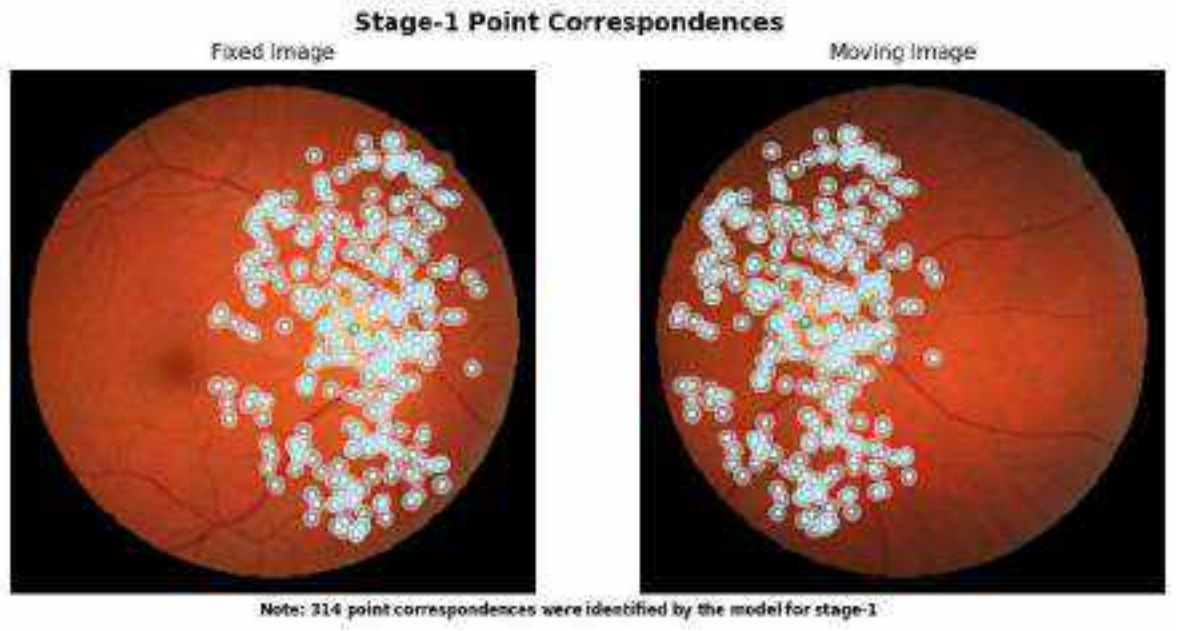

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



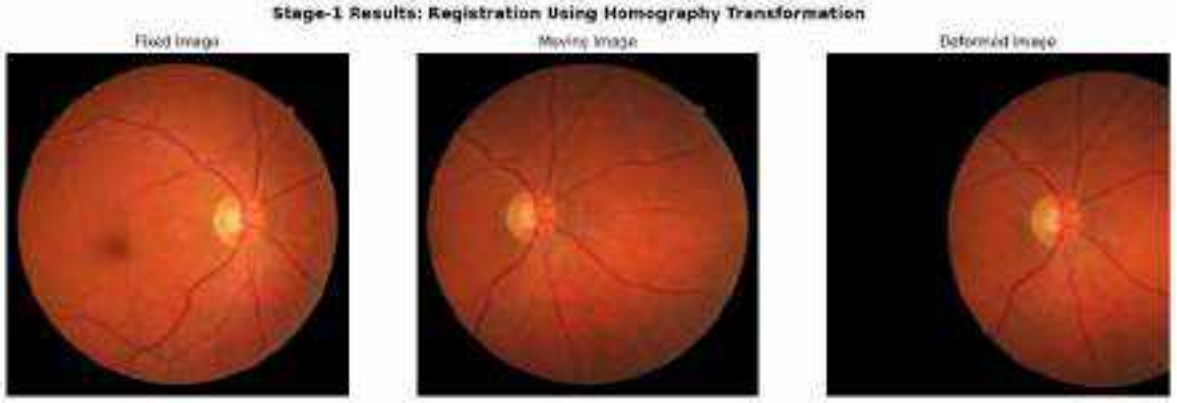
Note: 594 point correspondences were identified by the model for stage-2



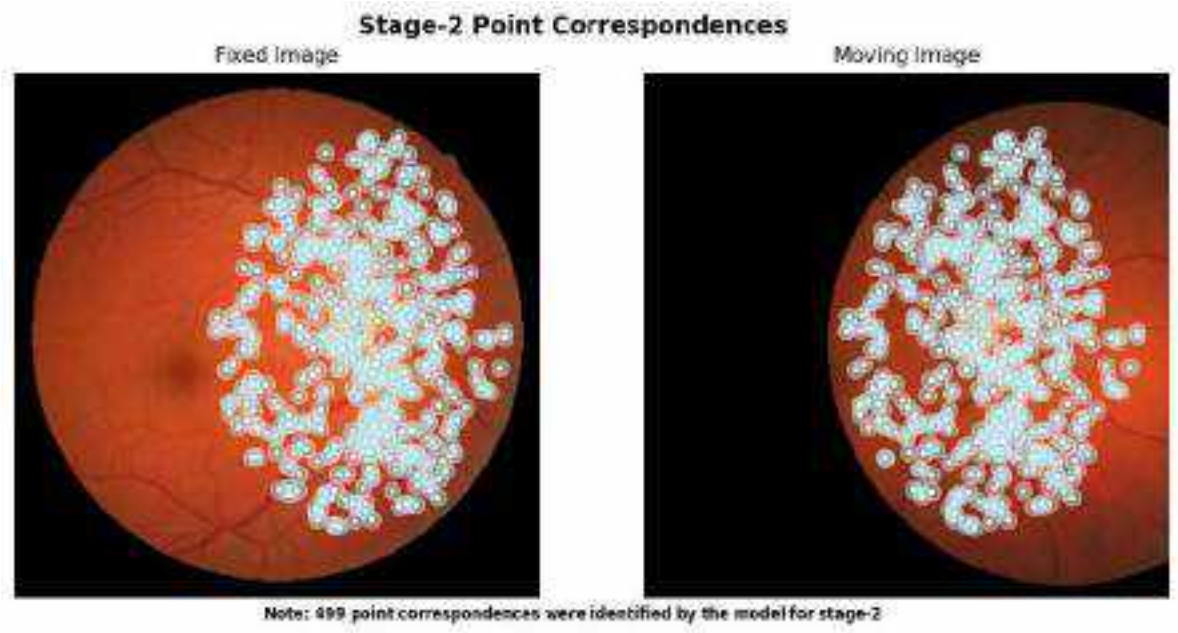
```
Mean Landmark Error for Case 6 Before Registration is 515.8562112023781 pixels
Mean Landmark Error for Case 6 After Registration is 4.223603712396644 pixels
Case 7
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P08_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P08_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

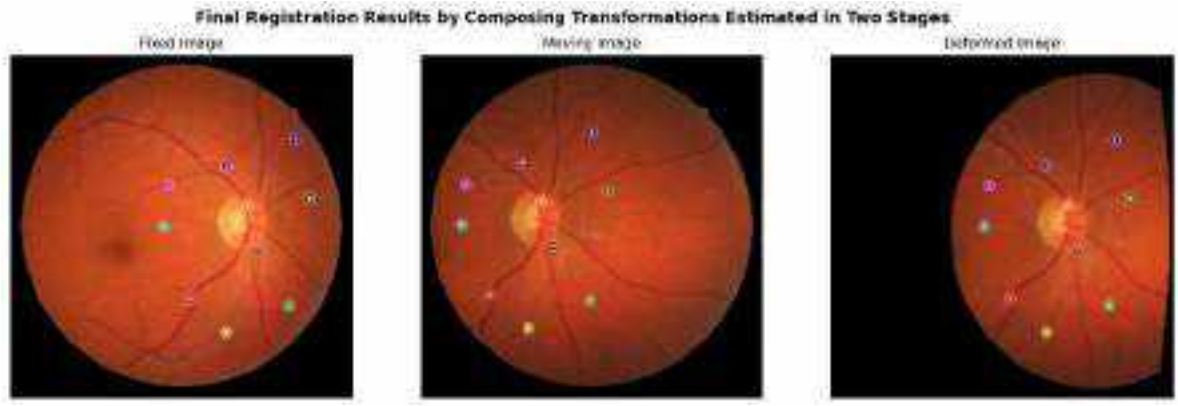



```
Homography Matrix:
[[ 1.13427446e+00 -2.77668276e-02  3.05050532e+02]
 [ 1.05748841e-01  1.05096321e+00 -2.22933367e+01]
 [ 1.46681591e-04  4.62041411e-06  1.00000000e+00]]
```

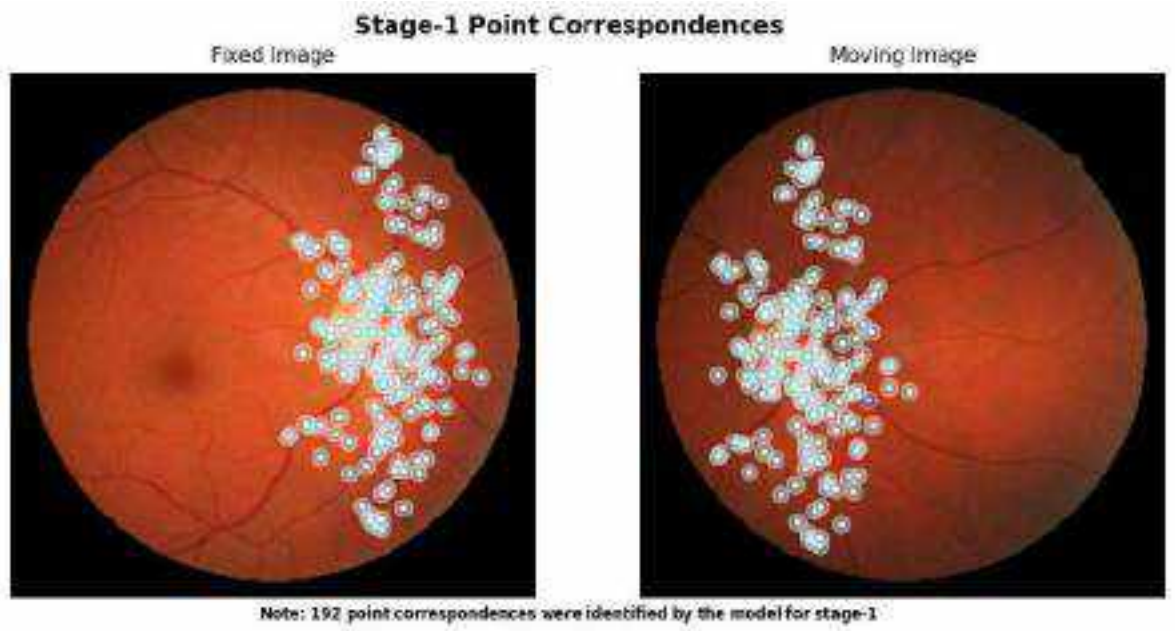



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



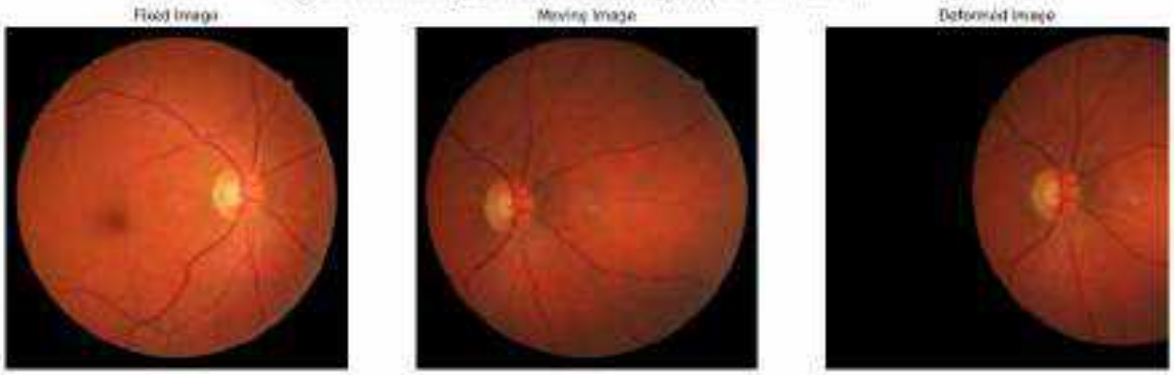



```
Mean Landmark Error for Case 7 Before Registration is 958.038245793841 pixels
Mean Landmark Error for Case 7 After Registration is 2.898212460788869 pixels
Case 8
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P09_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P09_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

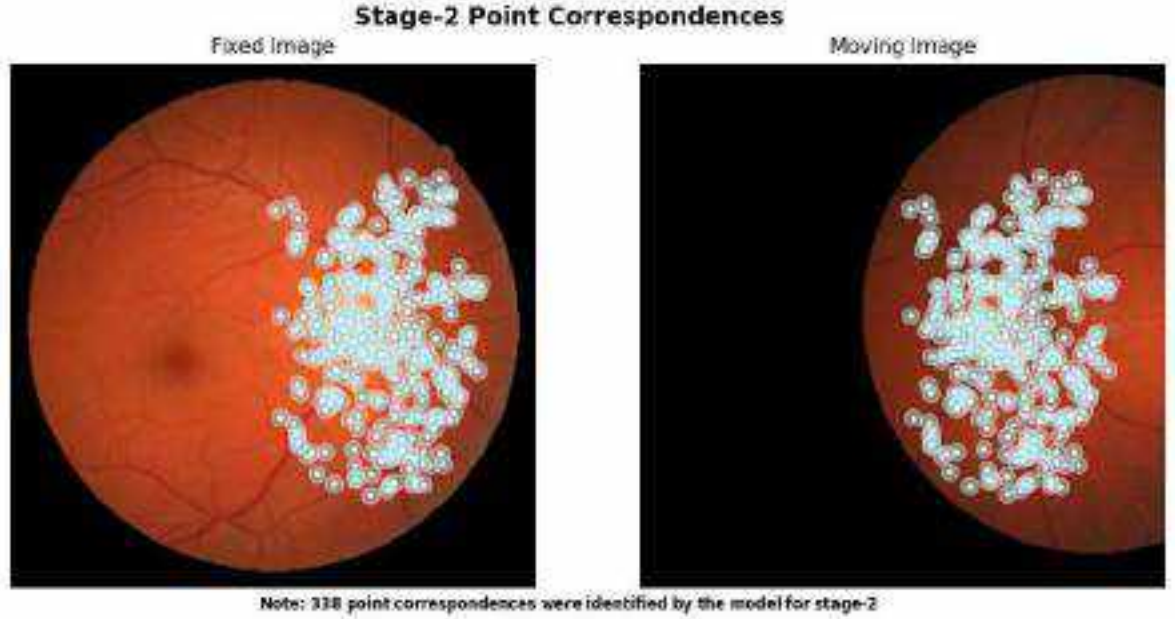


Note: 192 point correspondences were identified by the model for stage-1

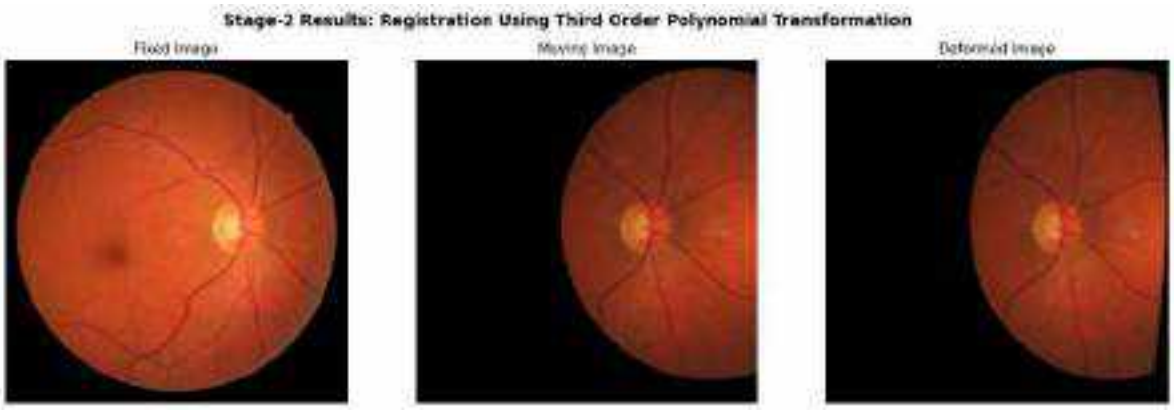
```
Homography Matrix:
[[ 1.10446055e+00 -5.90766711e-02  3.82631125e+02]
 [ 1.10184309e-01  1.01272768e+00 -5.64224283e+01]
 [ 1.53931867e-04 -3.09353651e-05  1.00000000e+00]]
```

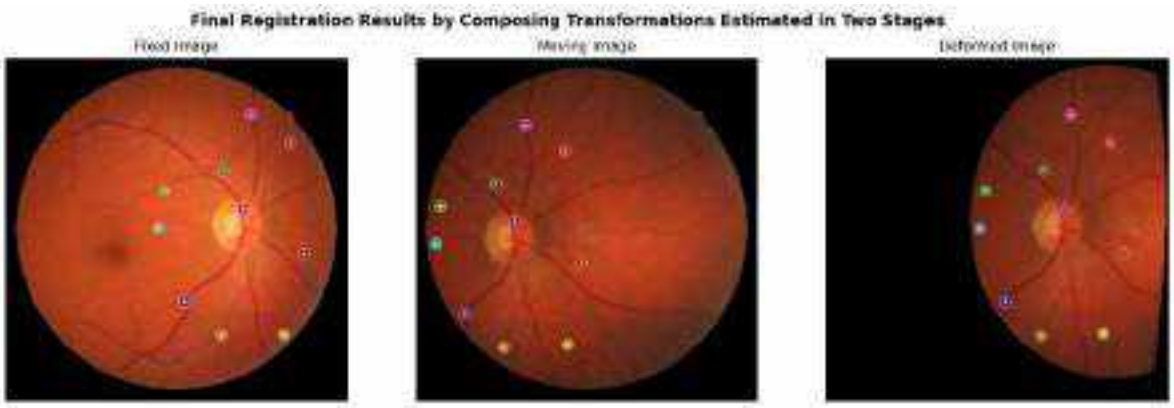



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

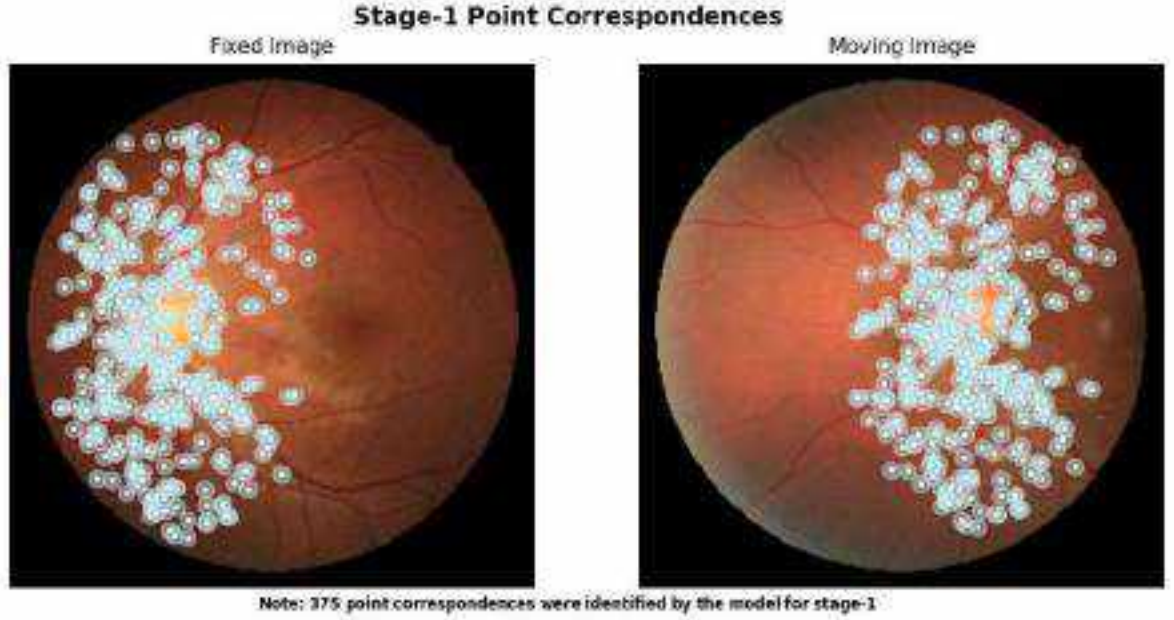


Note: 338 point correspondences were identified by the model for stage-2

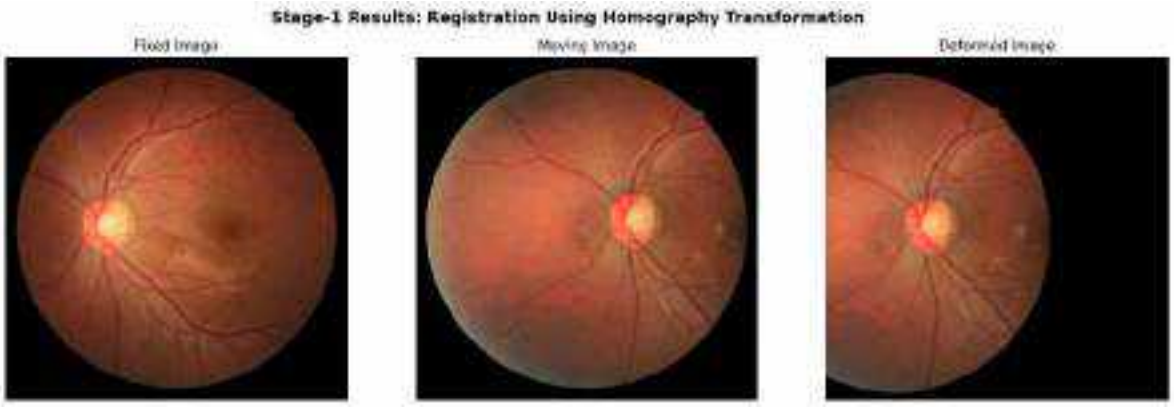




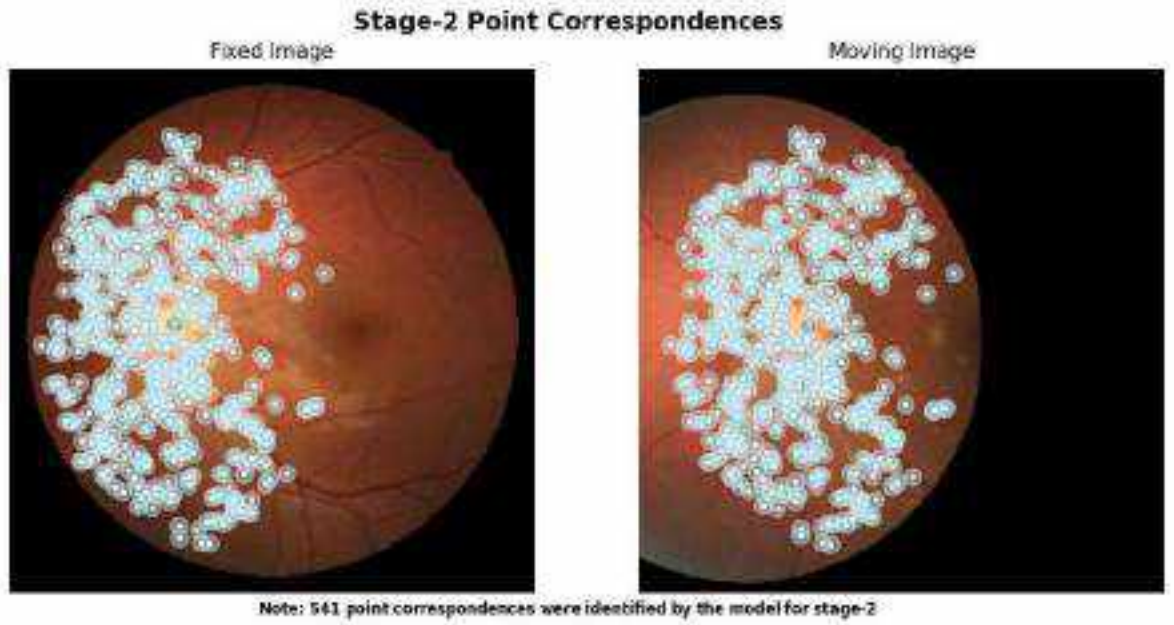

```
Mean Landmark Error for Case 8 Before Registration is 1137.4833111005107 pixels
Mean Landmark Error for Case 8 After Registration is 6.803350112491645 pixels
Case 9
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P10_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P10_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```




```
Homography Matrix:
[[ 8.83014185e-01  3.65049389e-02 -2.78870960e+02]
 [-1.02708116e-01  9.35262081e-01  6.45130984e+01]
 [-1.55448956e-04  1.70683696e-05  1.00000000e+00]]
```

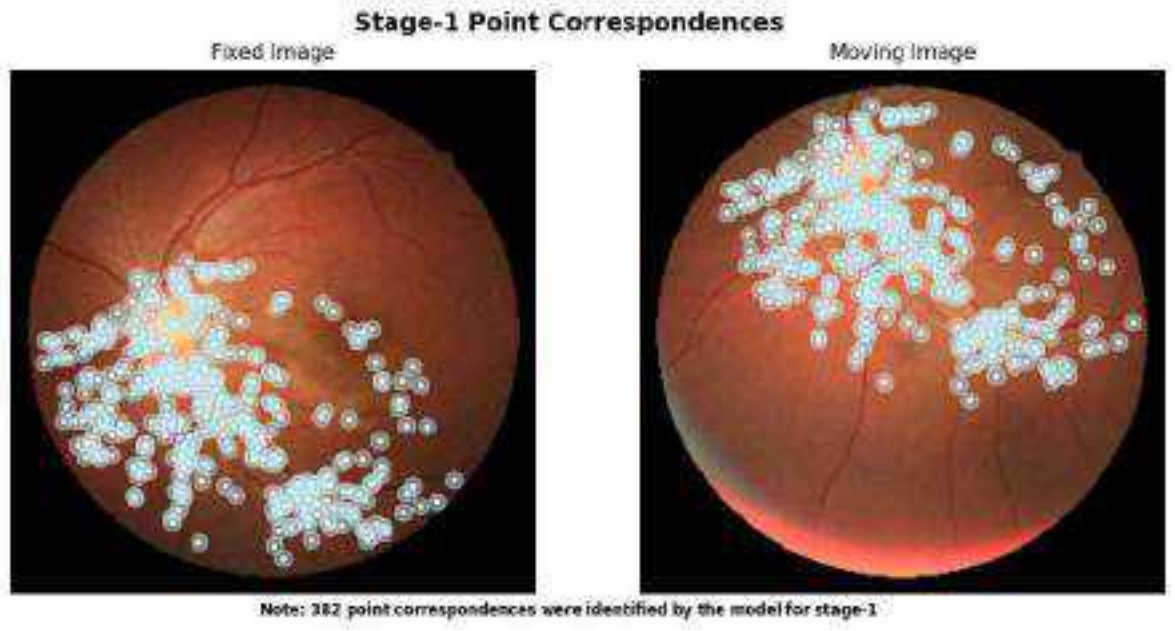



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



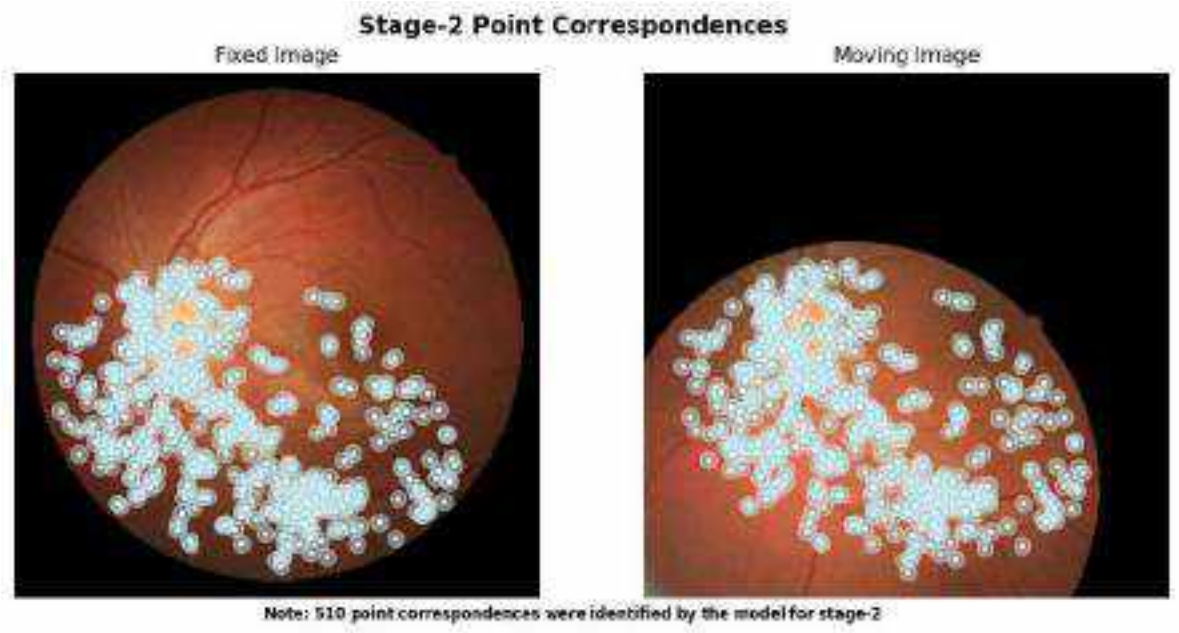
Note: 561 point correspondences were identified by the model for stage-2

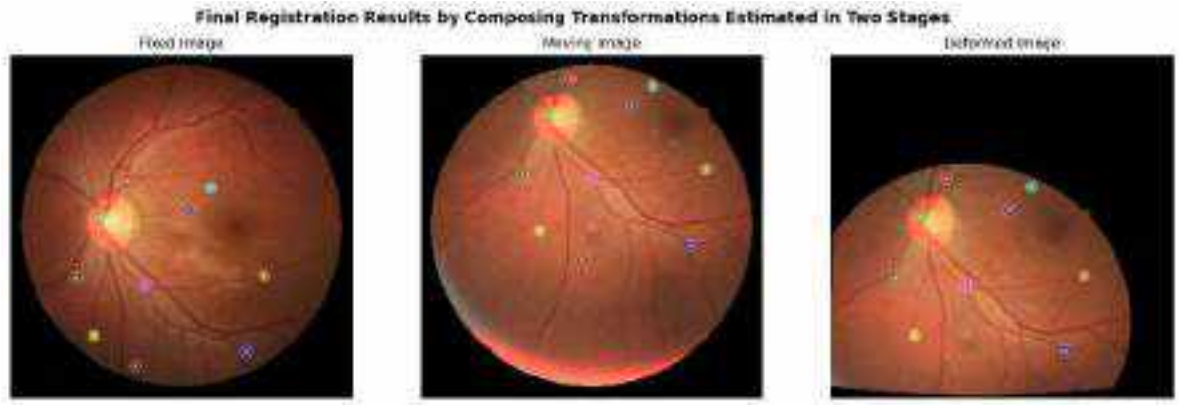
```
Mean Landmark Error for Case 9 Before Registration is 956.4652215482081 pixels
Mean Landmark Error for Case 9 After Registration is 3.60359374901644 pixels
Case 10
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P11_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P11_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



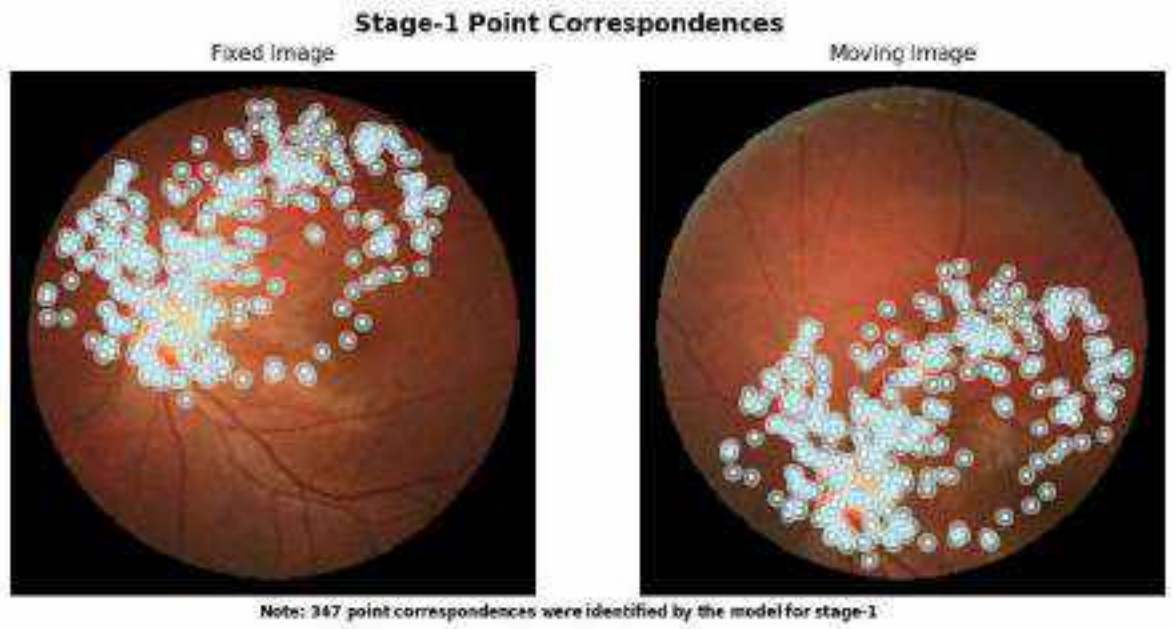

```
Homography Matrix:
[[ 1.02305643e+00  4.90005921e-02 -1.11705096e+02]
 [-7.80852484e-03  1.13709553e+00  2.62923746e+02]
 [-4.44260697e-05  1.54772373e-04  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



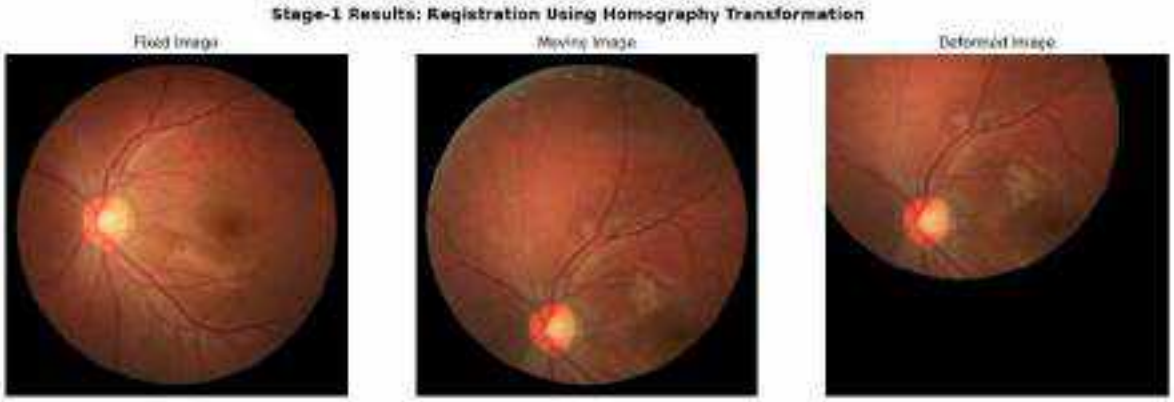



```
Mean Landmark Error for Case 10 Before Registration is 934.1023746683659 pixels
Mean Landmark Error for Case 10 After Registration is 4.523455531949551 pixels
Case 11
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P12_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P12_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

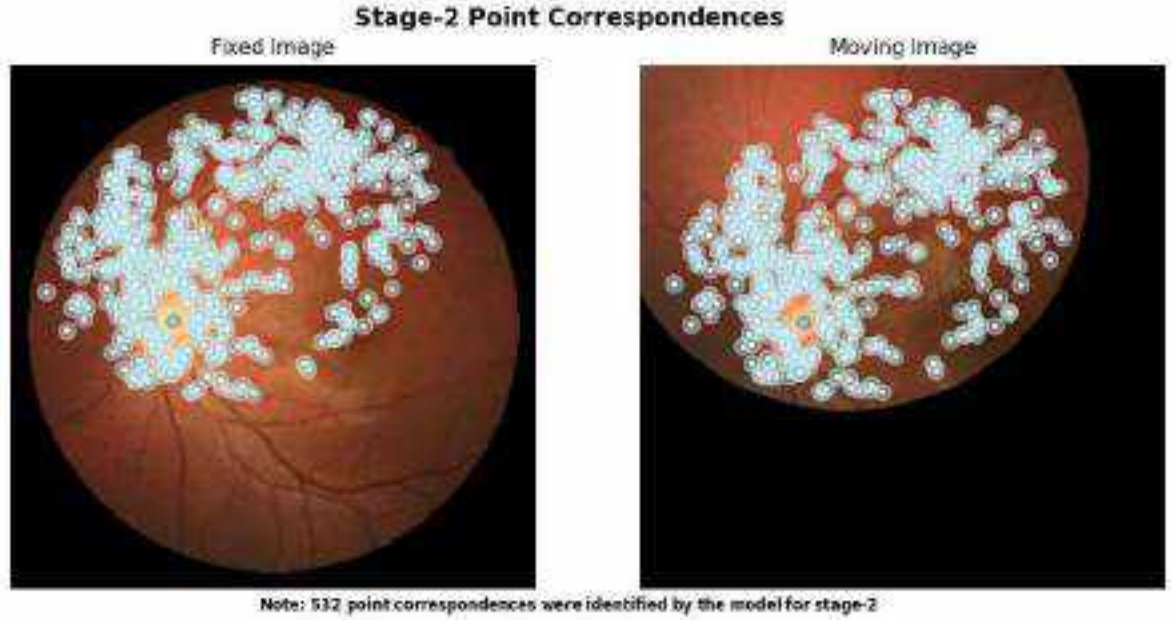


Note: 347 point correspondences were identified by the model for stage-1

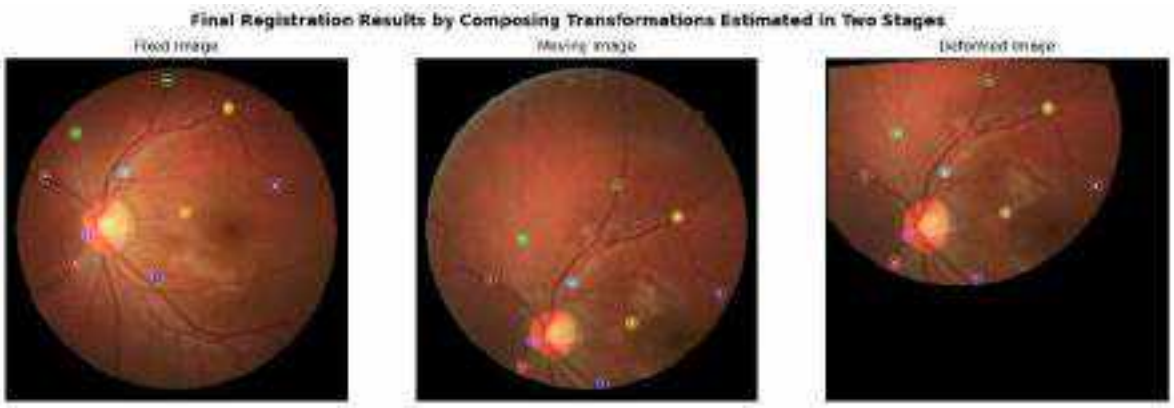
```
Homography Matrix:
[[ 8.81698373e-01 -3.52547974e-02 -6.05455281e+01]
 [-3.17003501e-02  8.50515668e-01 -2.17922961e+02]
 [-4.56291408e-05 -1.29618209e-04  1.00000000e+00]]
```



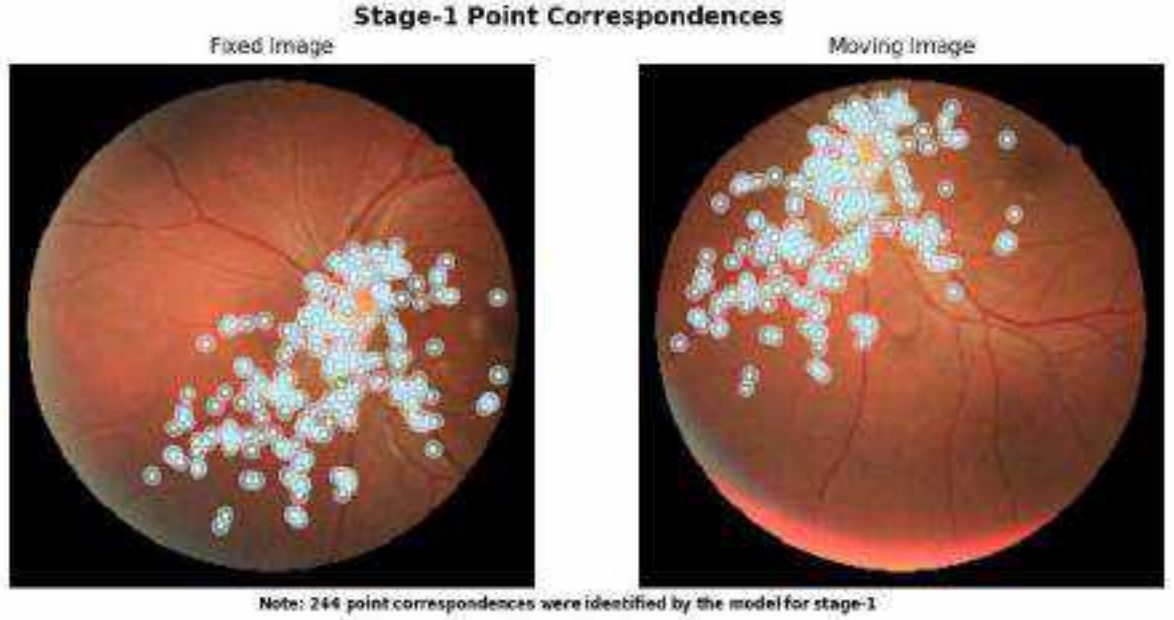

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



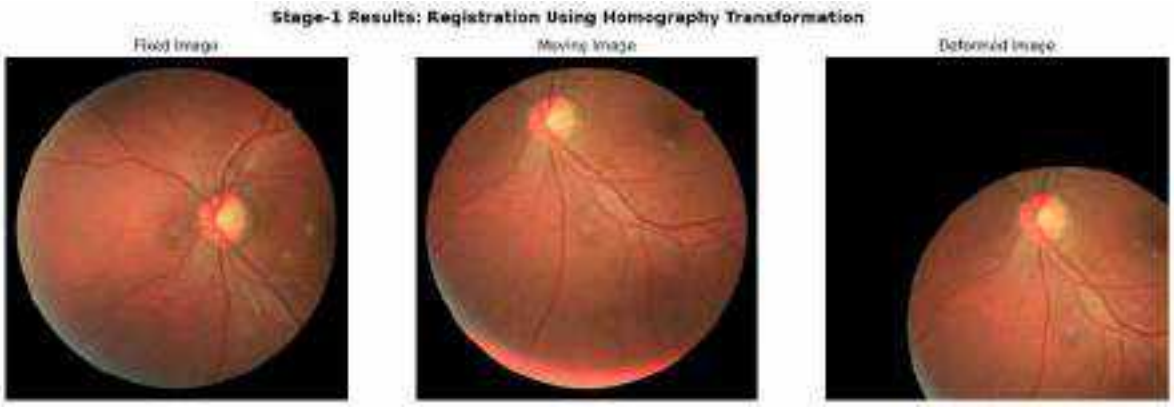
Note: 532 point correspondences were identified by the model for stage-2



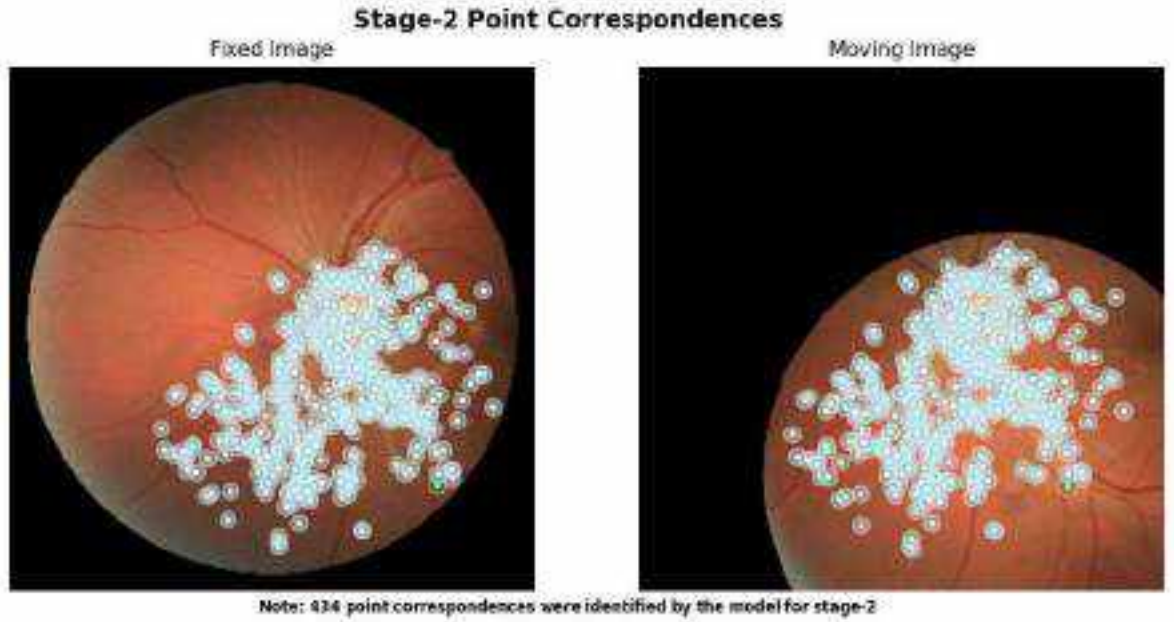

```
Mean Landmark Error for Case 11 Before Registration is 958.5424519633605 pixels
Mean Landmark Error for Case 11 After Registration is 4.171989577172561 pixels
Case 12
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P13_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P13_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



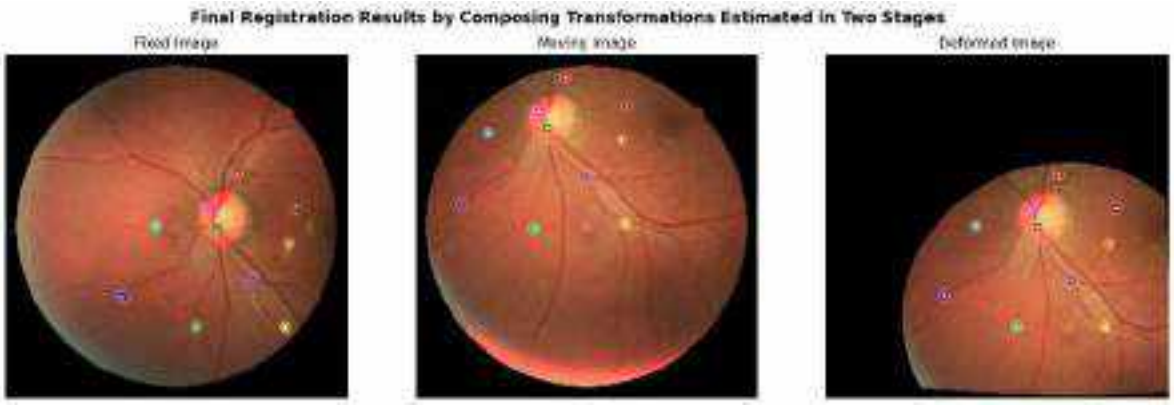

```
Homography Matrix:
[[1.16778021e+00 1.84606649e-03 1.97122812e+02]
 [1.66034701e-01 1.18278400e+00 2.04098597e+02]
 [1.35374239e-04 1.41060981e-04 1.00000000e+00]]
```



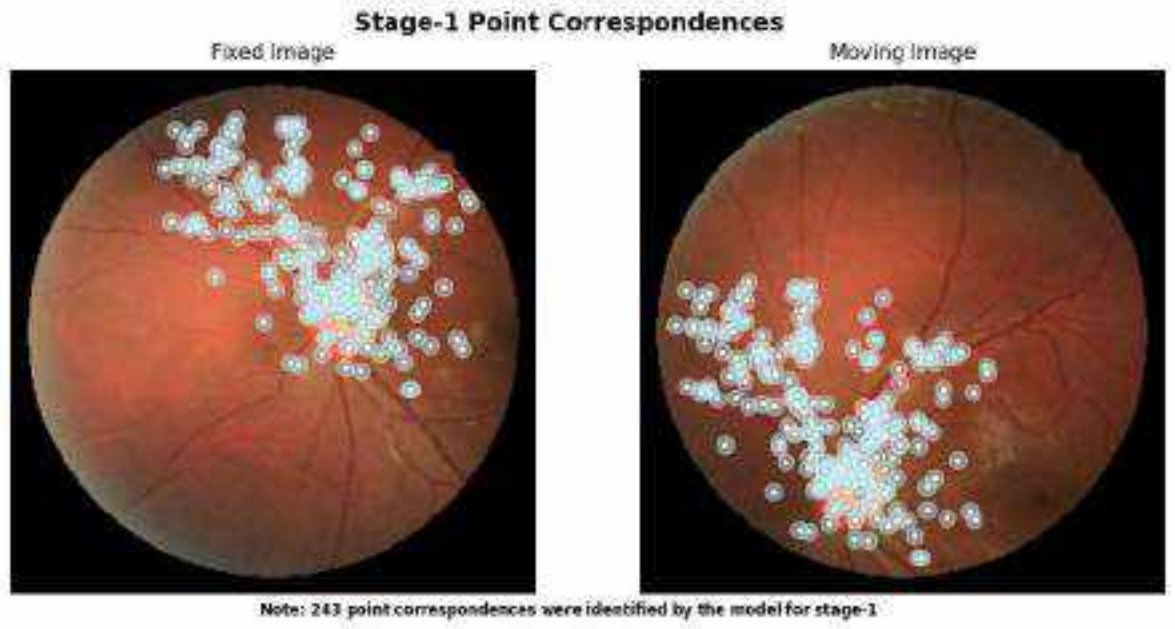

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



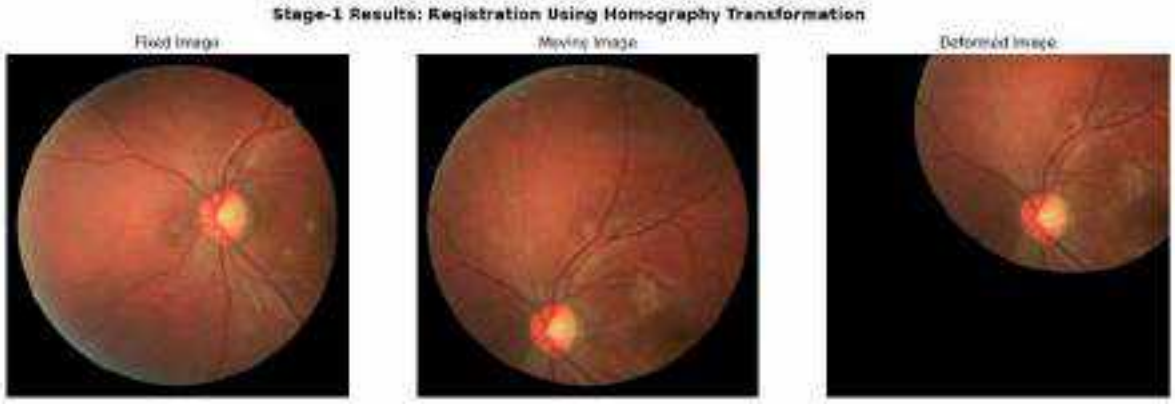
Note: 434 point correspondences were identified by the model for stage-2



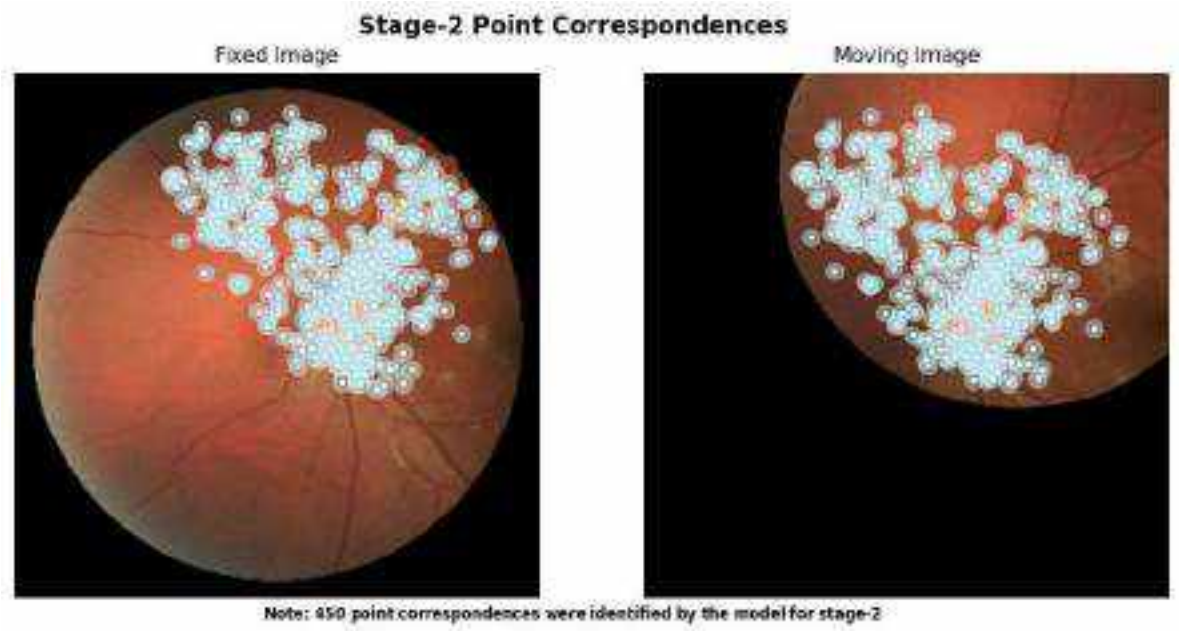
```
Mean Landmark Error for Case 12 Before Registration is 1064.243244891791 pixels
Mean Landmark Error for Case 12 After Registration is 4.930747662941249 pixels
Case 13
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P14_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P14_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

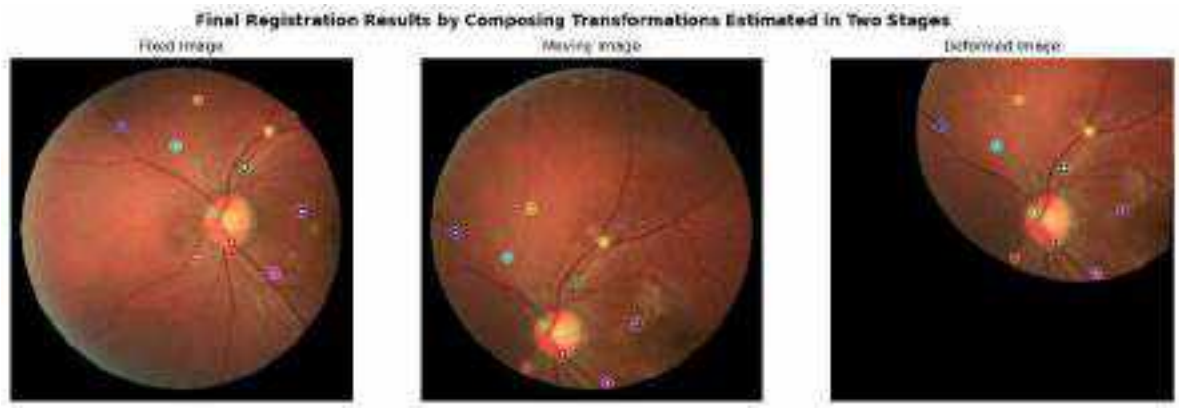



```
Homography Matrix:
[[ 1.00279550e+00 -6.78100405e-02  2.20797009e+02]
 [ 1.67109720e-02  8.67705136e-01 -2.43201597e+02]
 [ 1.20841666e-04 -1.58969369e-04  1.00000000e+00]]
```

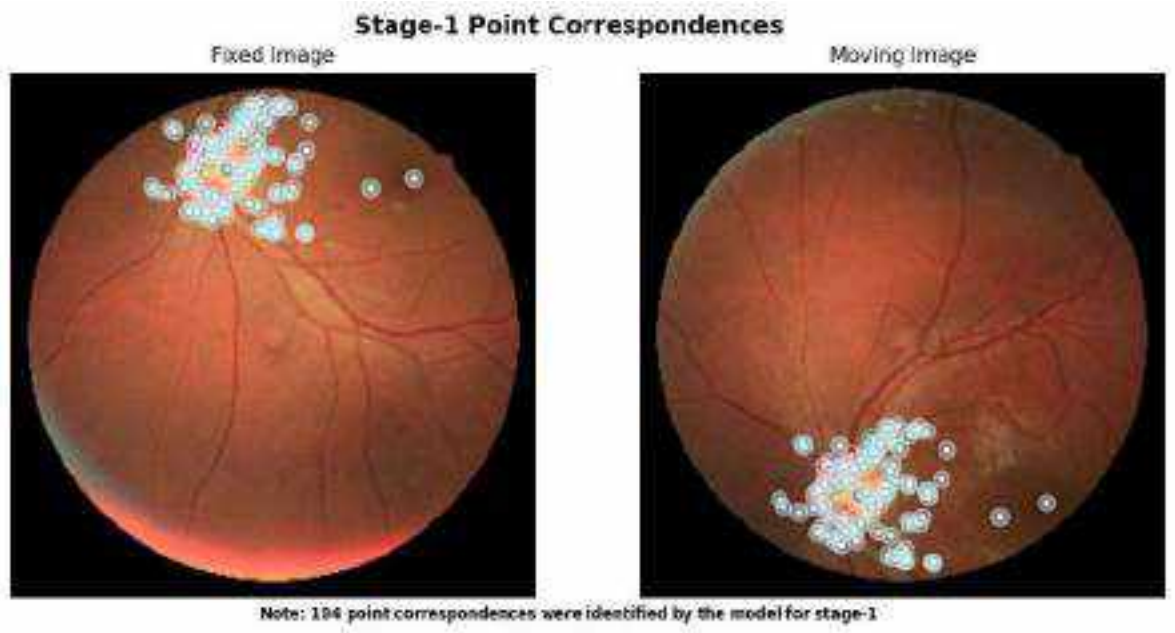



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

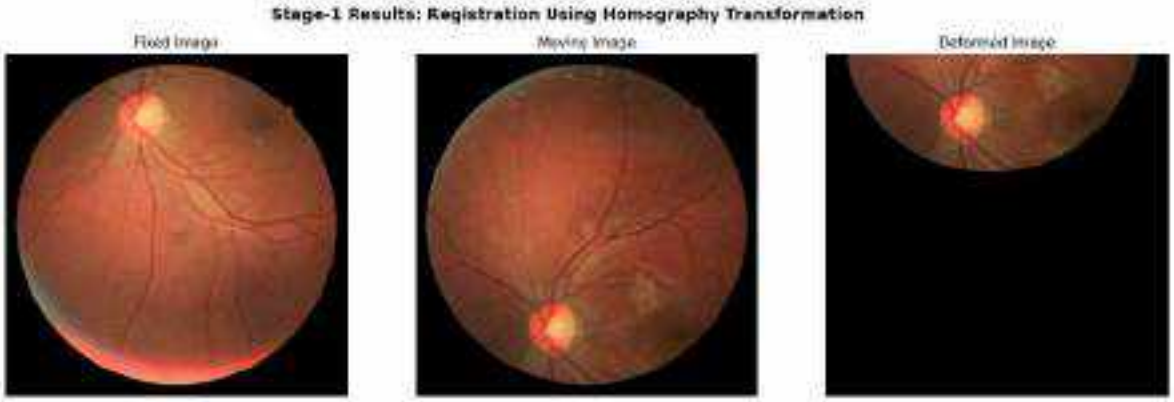




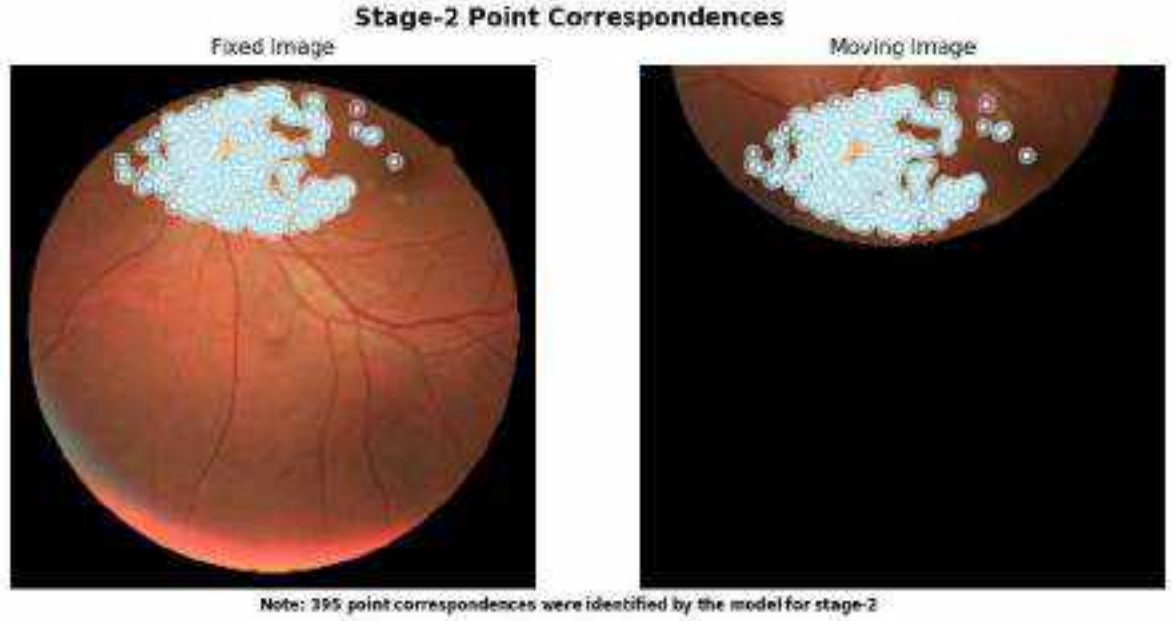

```
Mean Landmark Error for Case 13 Before Registration is 1149.237165590004 pixels
Mean Landmark Error for Case 13 After Registration is 5.716295686934598 pixels
Case 14
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P15_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P15_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

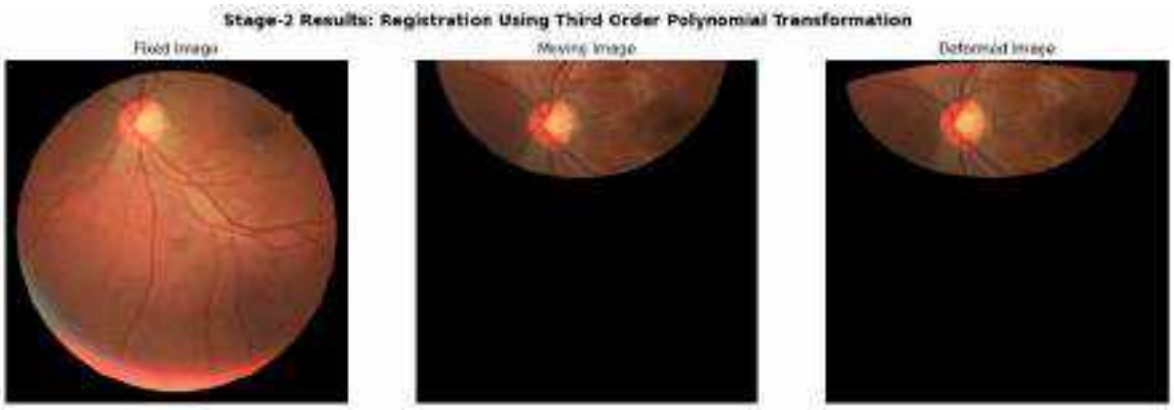


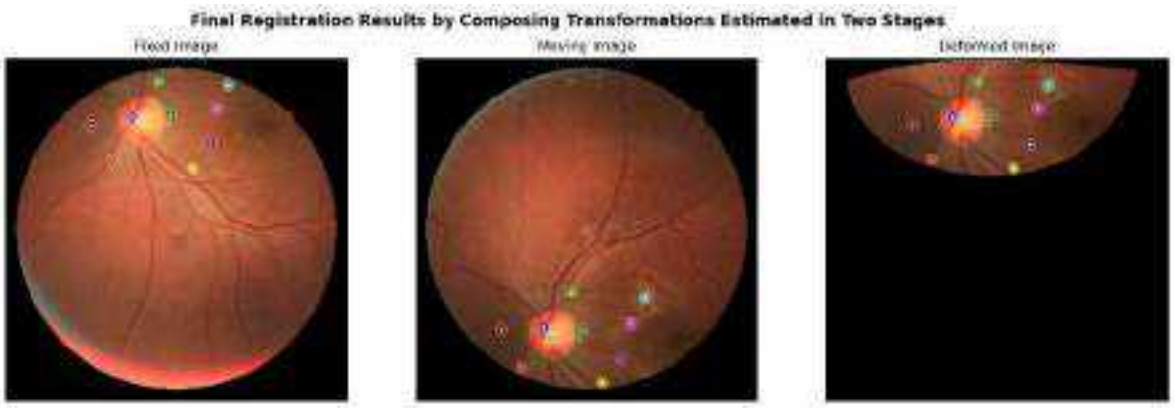

```
Homography Matrix:
[[ 8.05711972e-01 -5.33155022e-02  4.84249954e+01]
 [-5.14242163e-02  7.74524871e-01 -4.10557341e+02]
 [-2.32271479e-05 -2.10818242e-04  1.00000000e+00]]
```



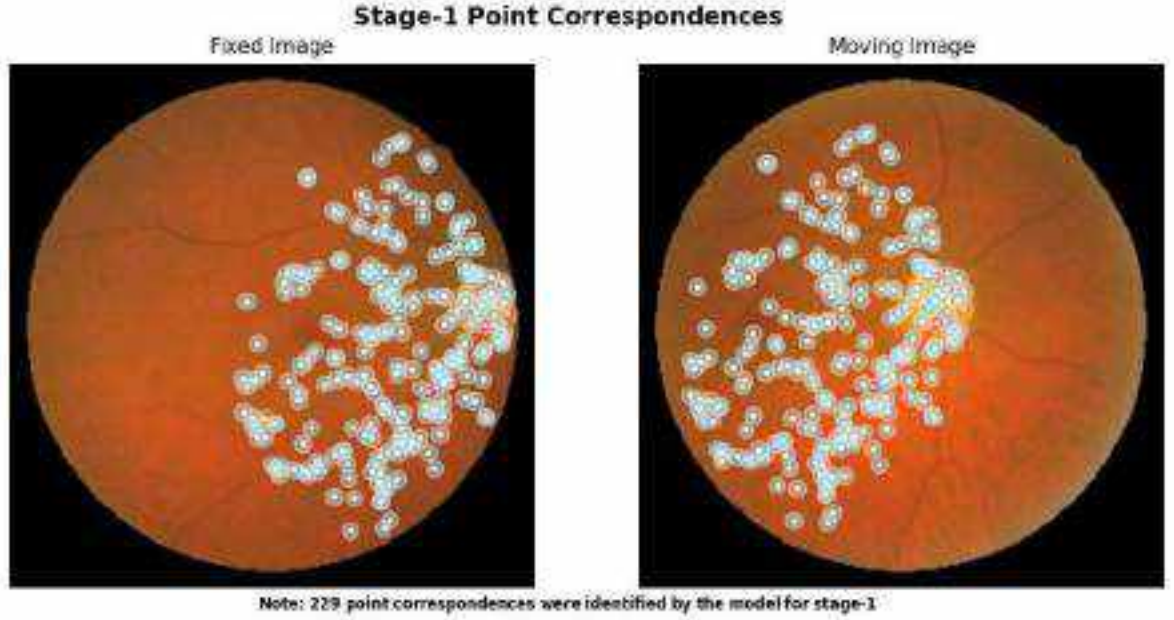

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



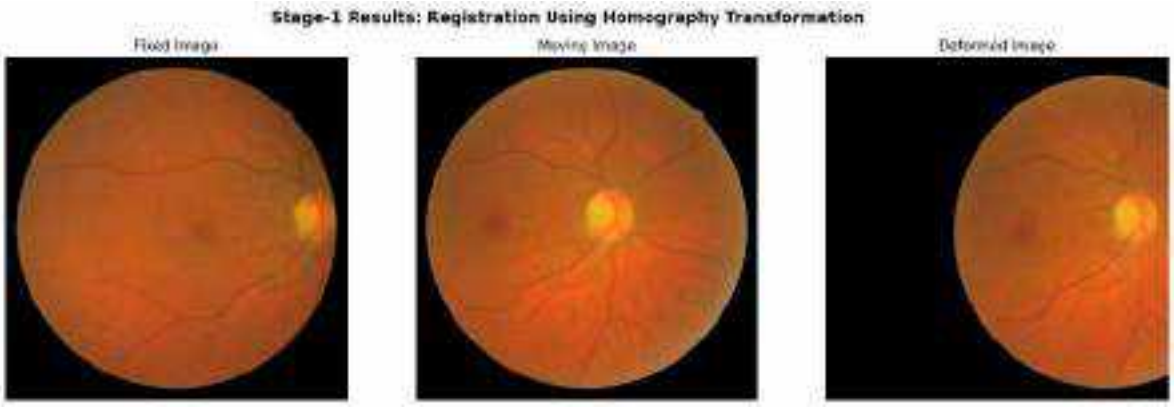




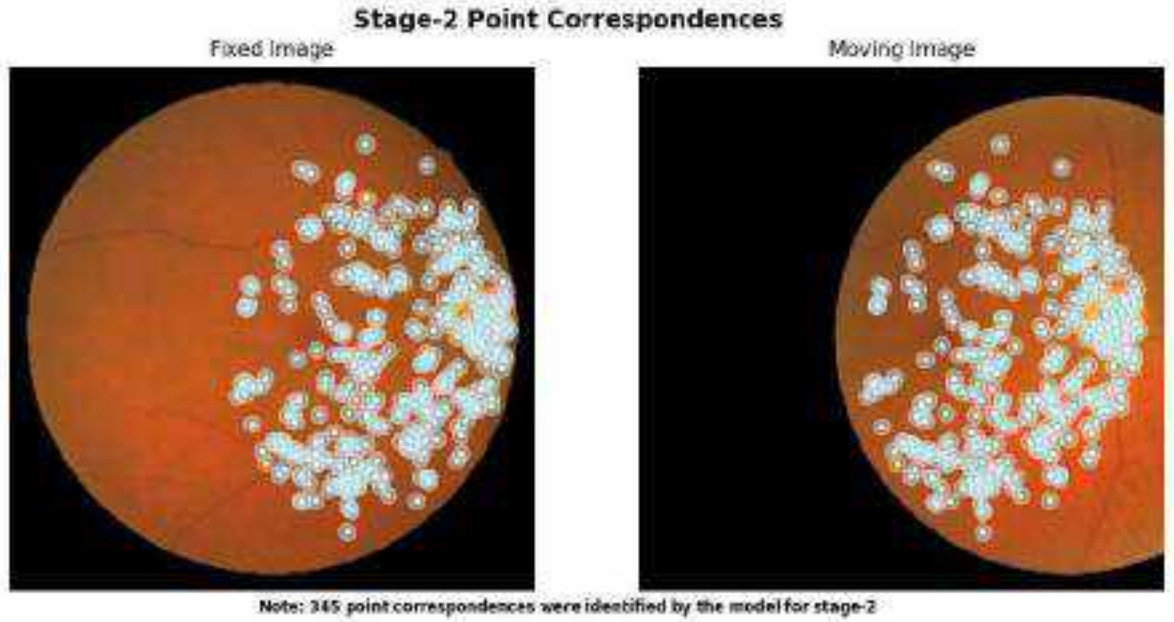

```
Mean Landmark Error for Case 14 Before Registration is 1796.826289802782 pixels
Mean Landmark Error for Case 14 After Registration is 3.9501586189877513 pixels
Case 15
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P16_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P16_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



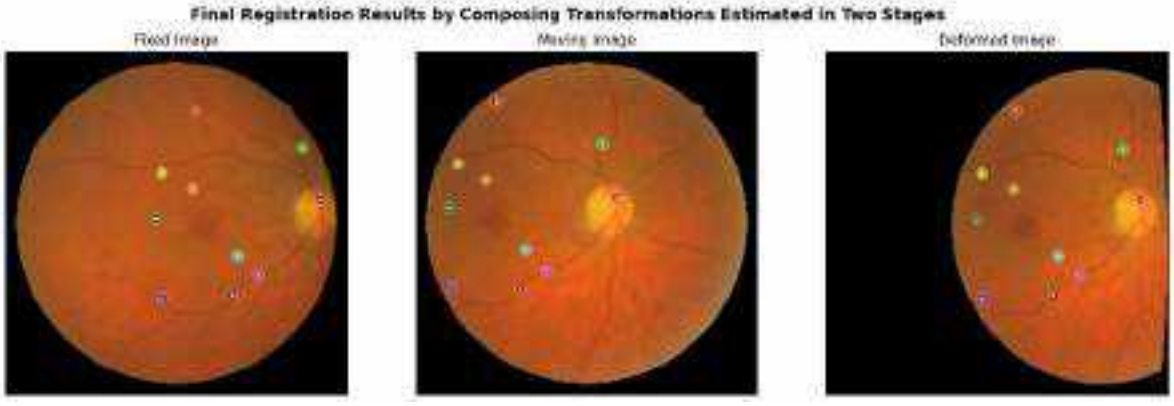

```
Homography Matrix:
[[ 1.10823316e+00  4.34577008e-02  2.91119337e+02]
 [ 9.20868763e-03  1.02452520e+00  2.01569014e+01]
 [ 1.34752889e-04 -7.91529007e-06  1.00000000e+00]]
```



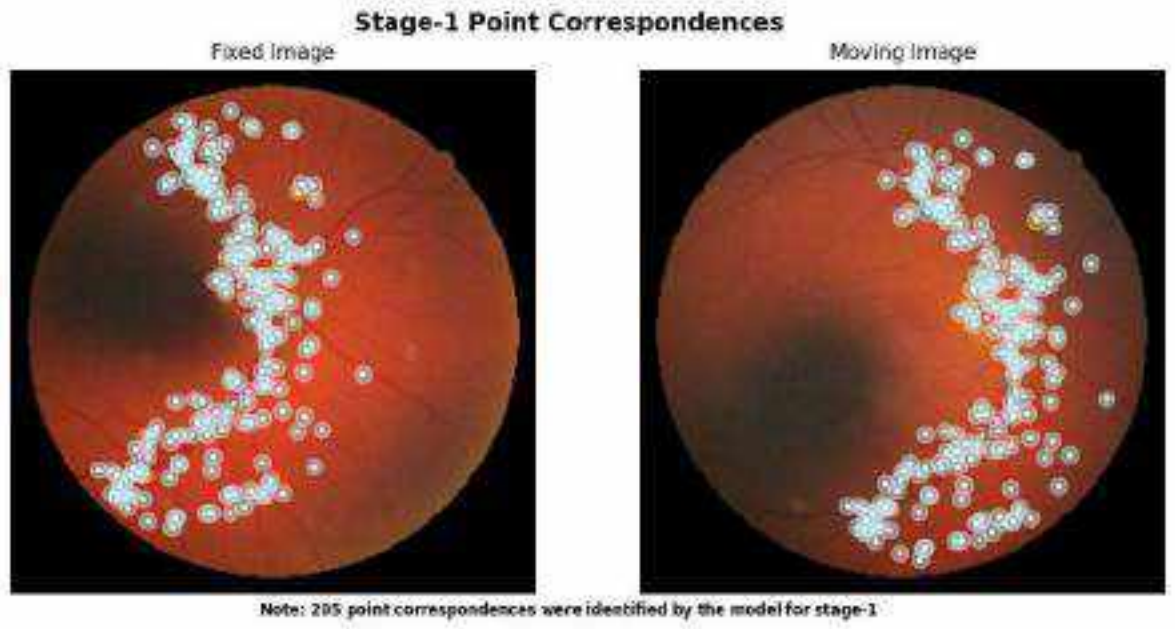

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



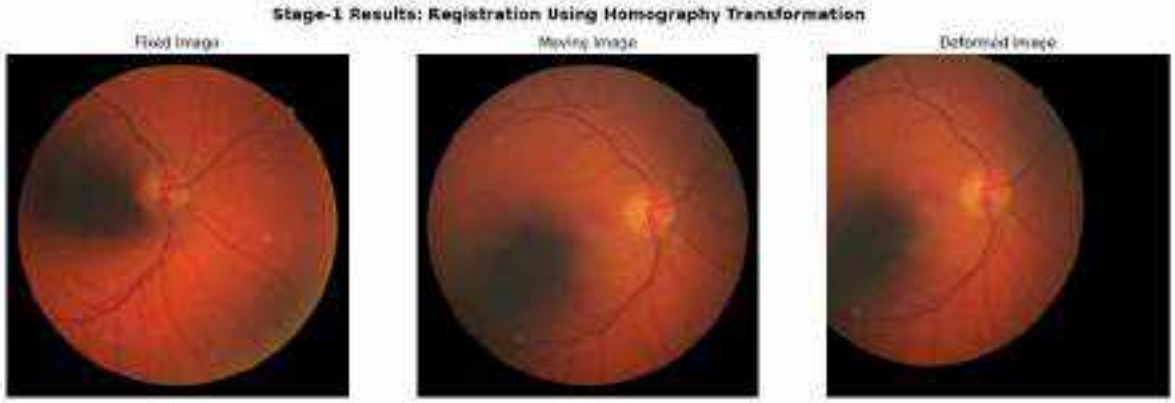
Note: 345 point correspondences were identified by the model for stage-2



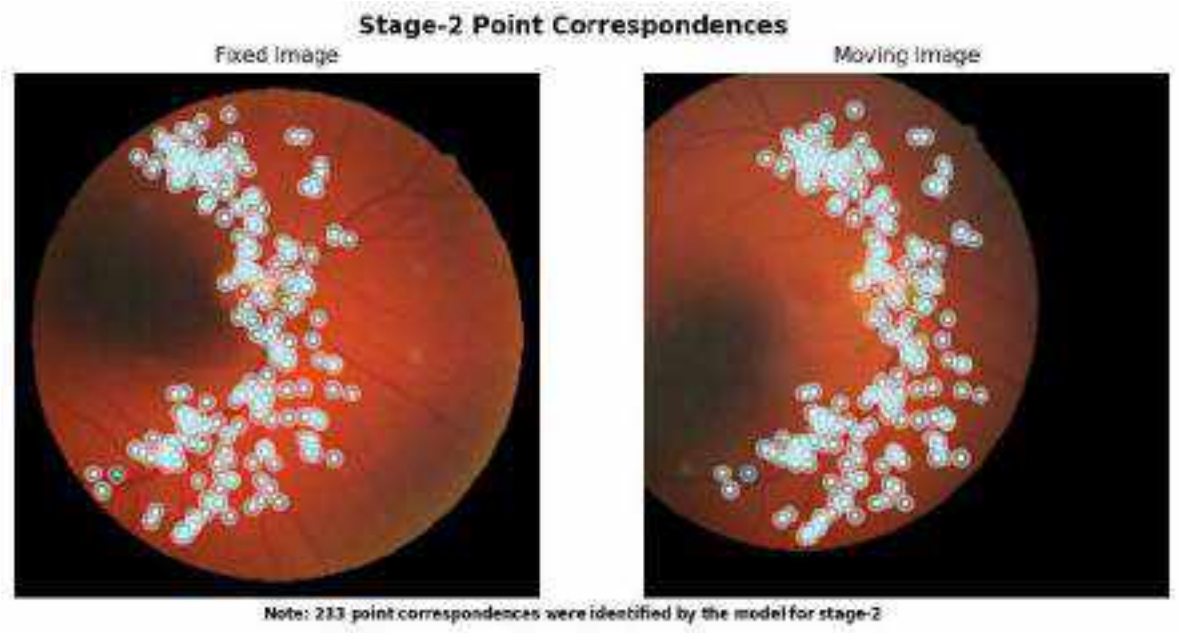
```
Mean Landmark Error for Case 15 Before Registration is 998.6574626707416 pixels
Mean Landmark Error for Case 15 After Registration is 3.909571730197989 pixels
Case 16
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P17_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P17_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

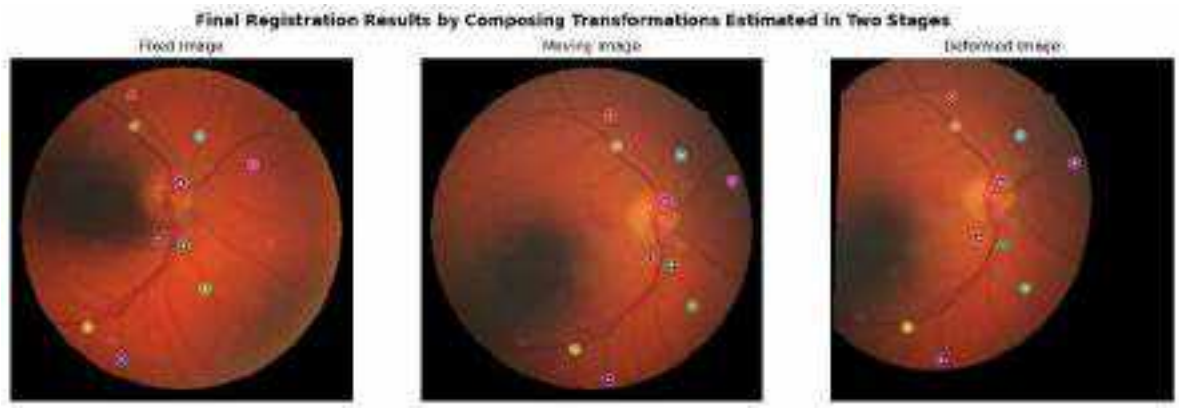



```
Homography Matrix:
[[ 8.60179631e-01 -5.13320714e-02 -1.27104059e+02]
 [-1.27986348e-02  9.03811997e-01 -3.57642584e+01]
 [-1.00651831e-04 -4.53931097e-05  1.00000000e+00]]
```

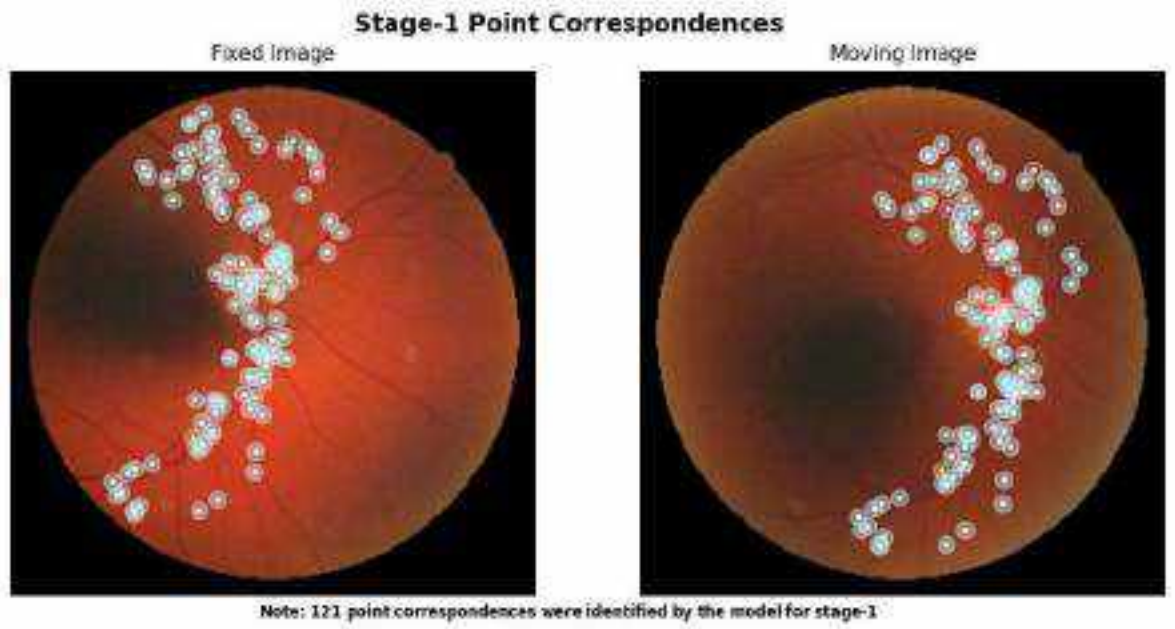



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



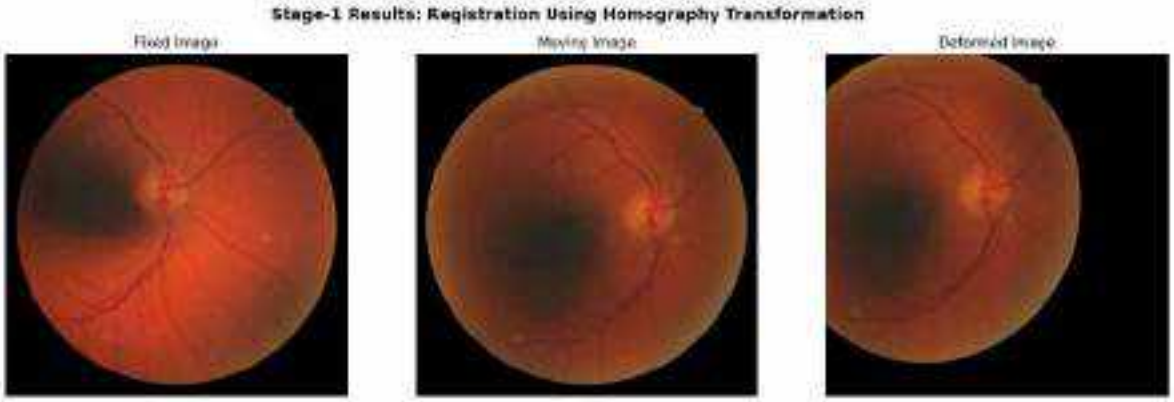



```
Mean Landmark Error for Case 16 Before Registration is 652.8864601525476 pixels
Mean Landmark Error for Case 16 After Registration is 5.540414471908641 pixels
Case 17
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P18_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P18_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

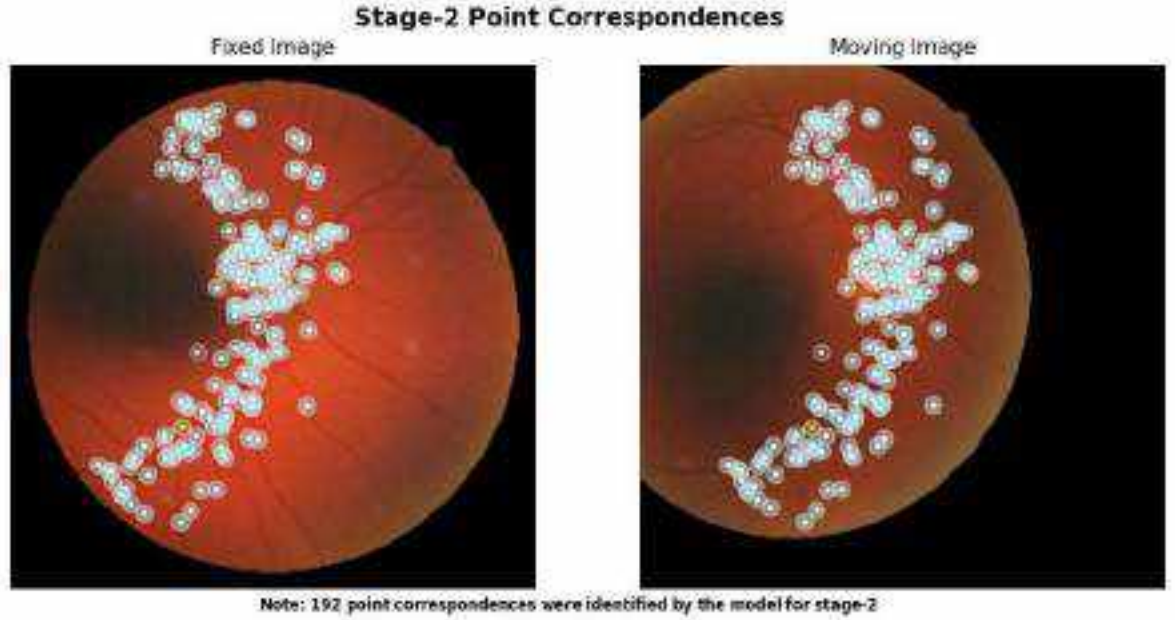


Note: 121 point correspondences were identified by the model for stage-1

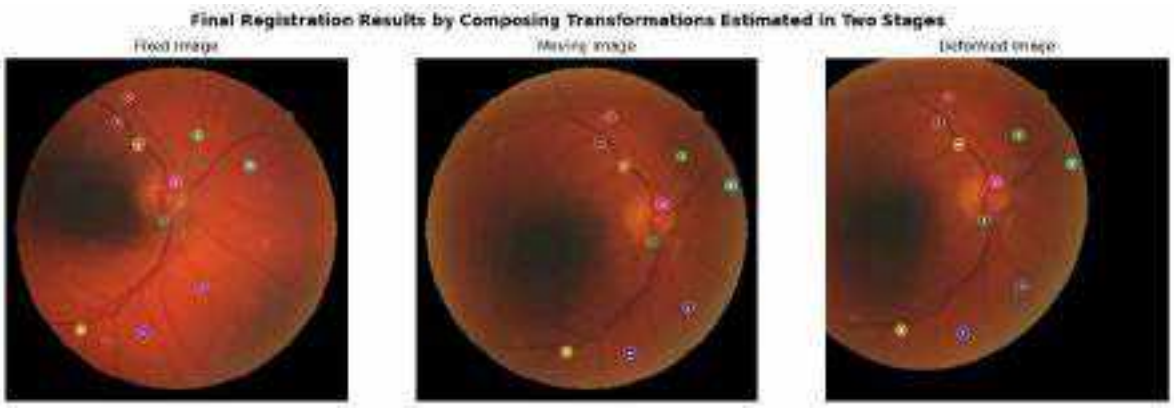
```
Homography Matrix:
[[ 8.71688665e-01 -3.45499523e-02 -1.39078903e+02]
 [-3.43679883e-02  9.23478365e-01 -2.99006263e+01]
 [-9.20619293e-05 -2.59273456e-05  1.00000000e+00]]
```



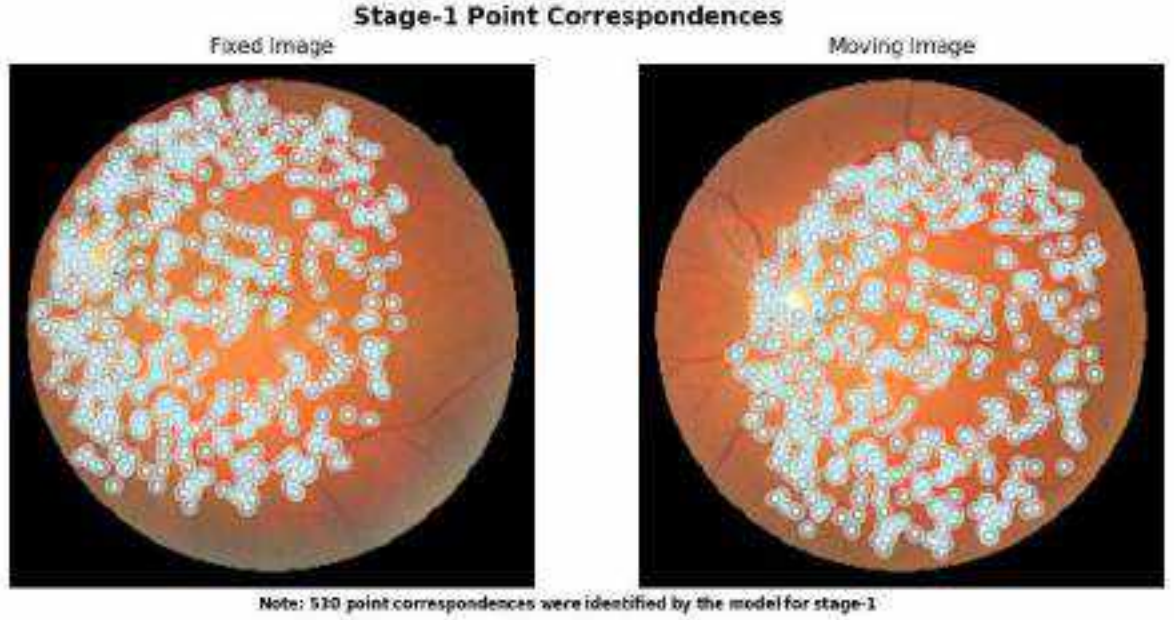

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



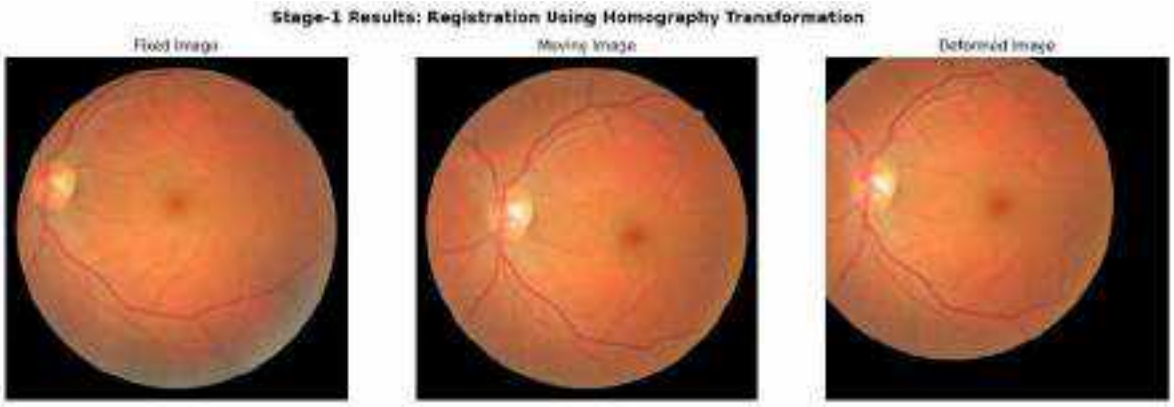
Note: 192 point correspondences were identified by the model for stage-2



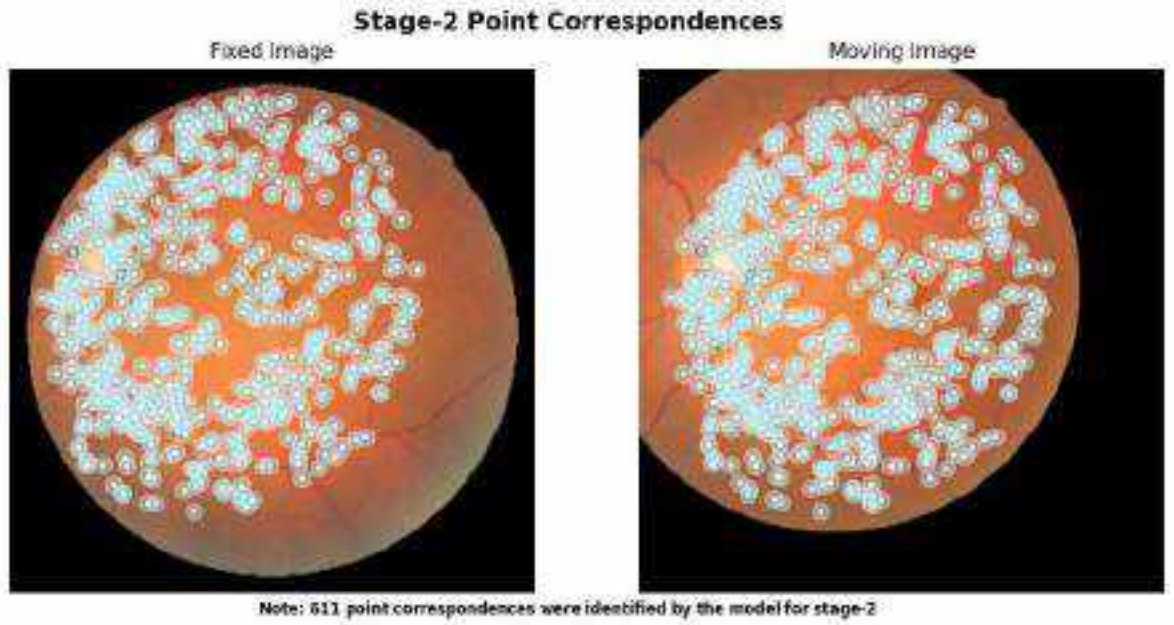

```
Mean Landmark Error for Case 17 Before Registration is 660.8728672766899 pixels
Mean Landmark Error for Case 17 After Registration is 5.346280486188029 5 pixels
Case 18
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P19_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P19_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



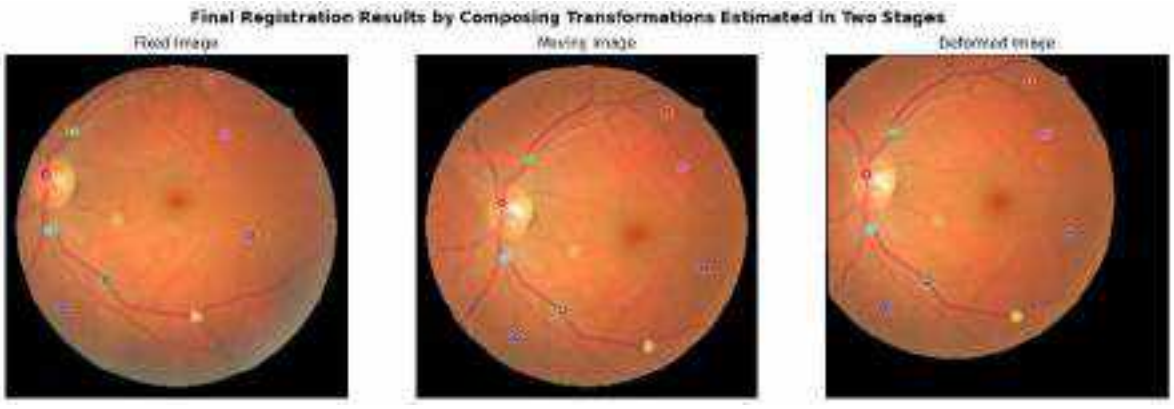

```
Homography Matrix:
[[ 9.42071200e-01  8.26083988e-03 -1.15742589e+02]
 [-4.42597946e-02  9.46842338e-01 -5.20724344e+01]
 [-4.96754431e-05 -3.38104271e-05  1.00000000e+00]]
```



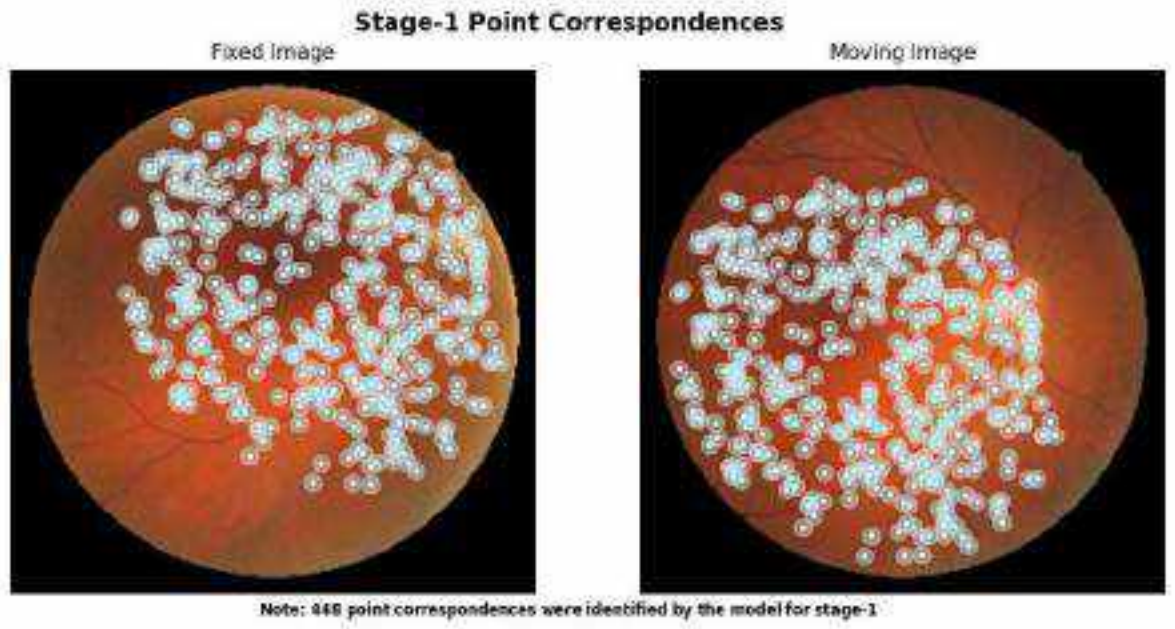

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



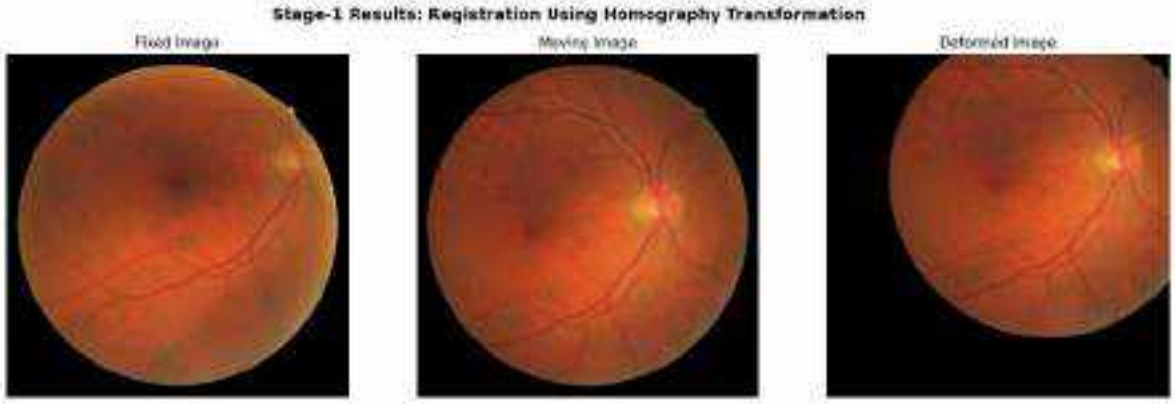
Note: 611 point correspondences were identified by the model for stage-2



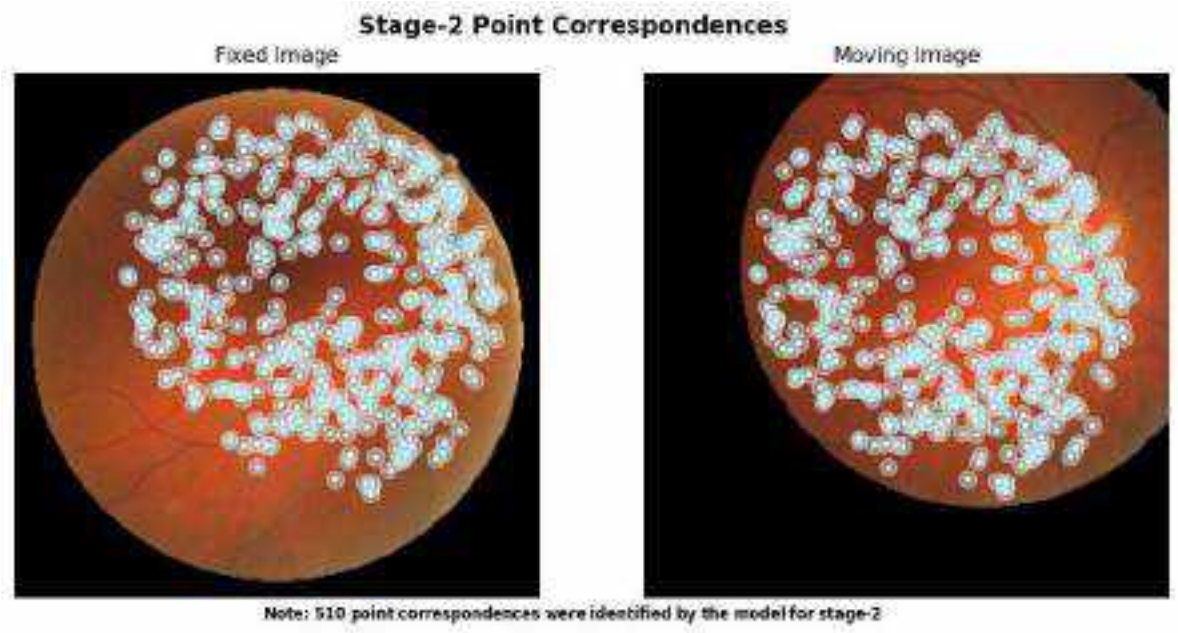
```
Mean Landmark Error for Case 18 Before Registration is 464.40406664315043 pixels
Mean Landmark Error for Case 18 After Registration is 3.3924516908368894 pixels
Case 19
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P20_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P20_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

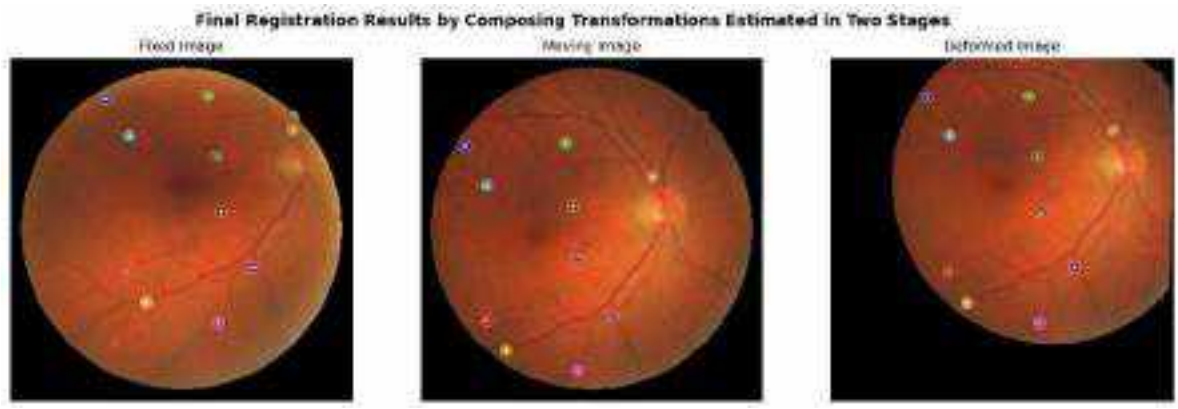



```
Homography Matrix:
[[ 1.01726350e+00 -3.99306793e-02  1.50857345e+02]
 [ 3.05069838e-02  9.54703203e-01 -1.26068430e+02]
 [ 6.09870640e-05 -6.58177453e-05  1.00000000e+00]]
```

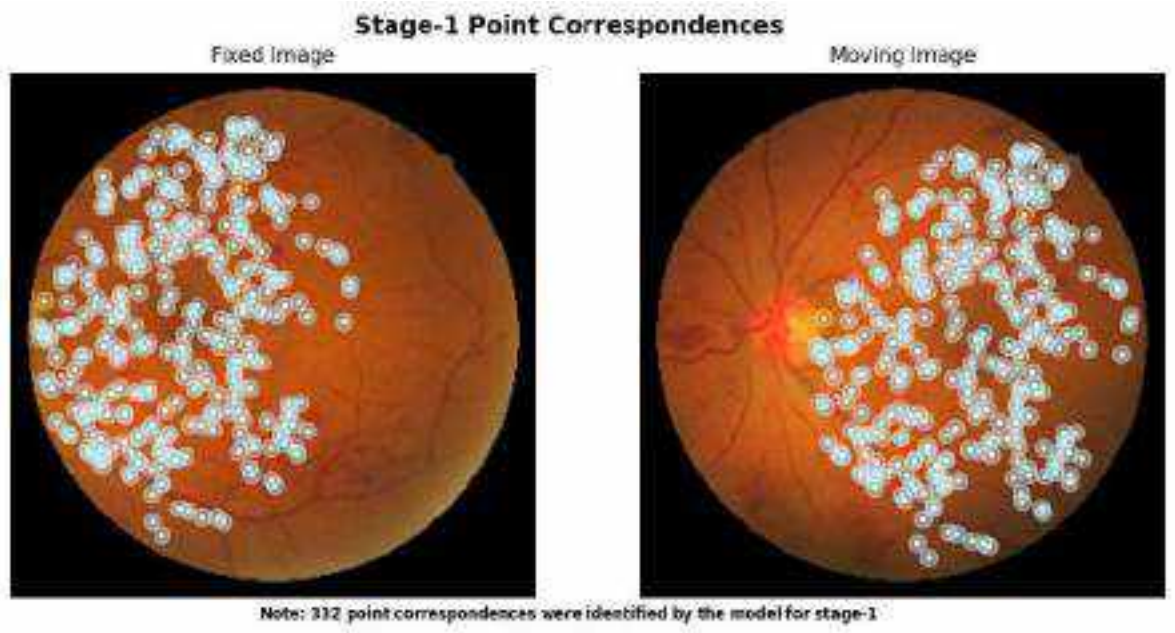



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





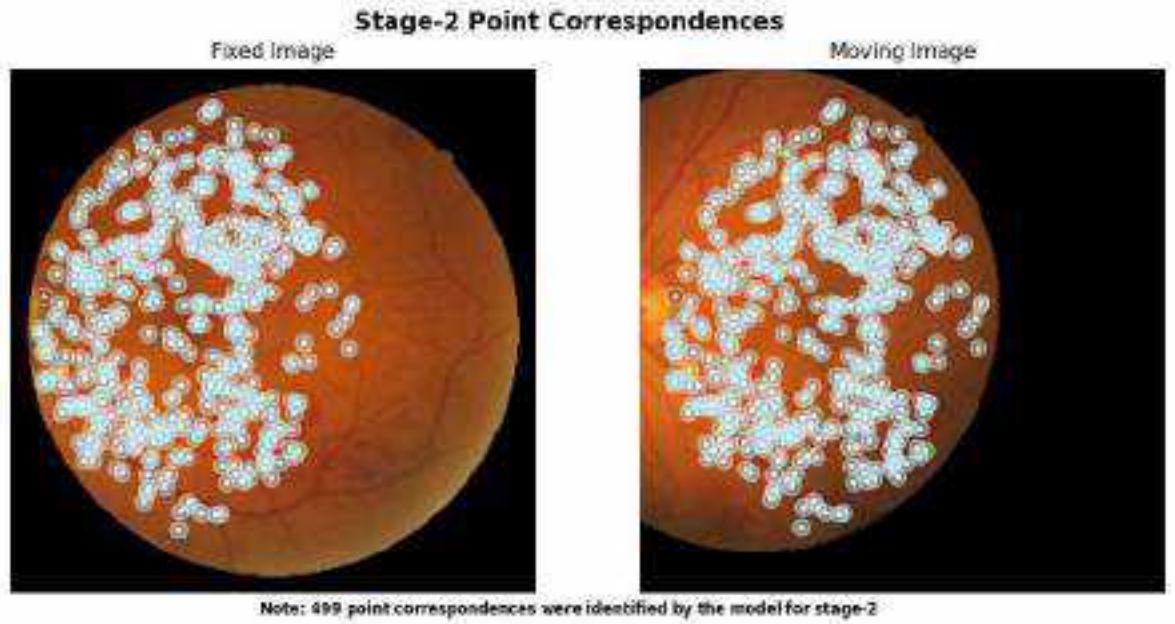

```
Mean Landmark Error for Case 19 Before Registration is 604.2822906840928 pixels
Mean Landmark Error for Case 19 After Registration is 3.2441584581077 61 pixels
Case 20
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P21_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P21_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



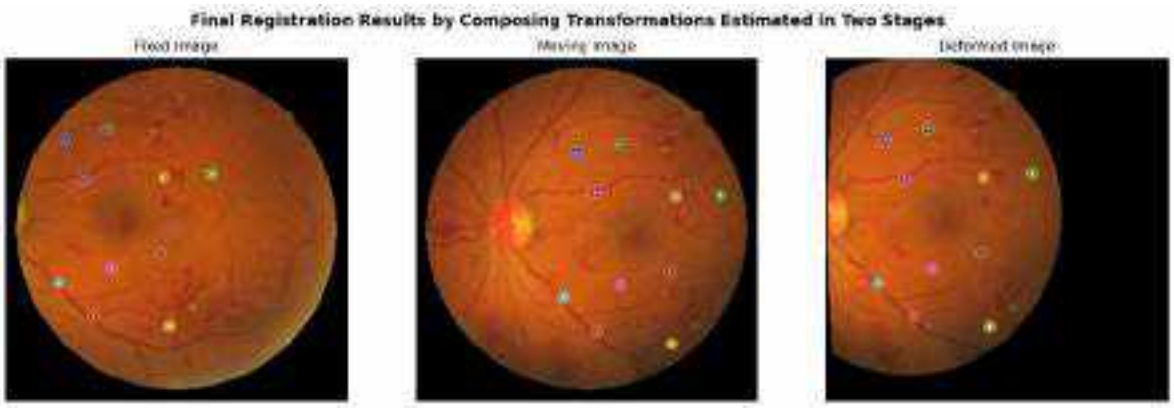
Note: 312 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.11200294e-01  3.28794669e-02 -2.48498487e+02]
 [-8.39356246e-02  9.33181076e-01  1.11007672e+01]
 [-8.42186532e-05 -1.62200157e-05  1.00000000e+00]]
```

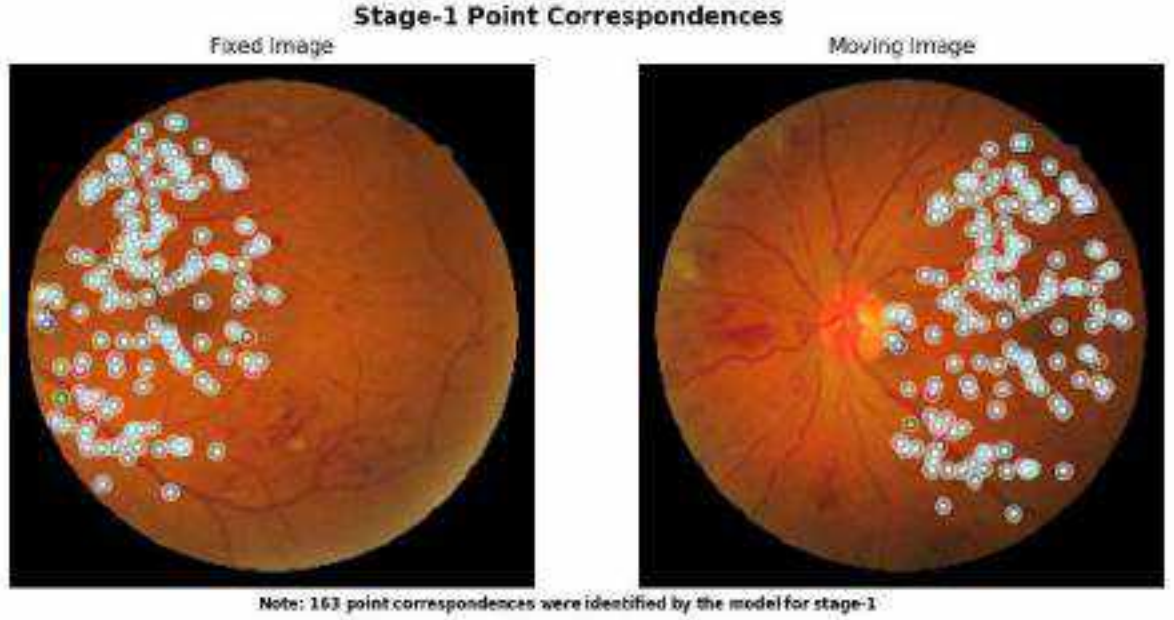

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



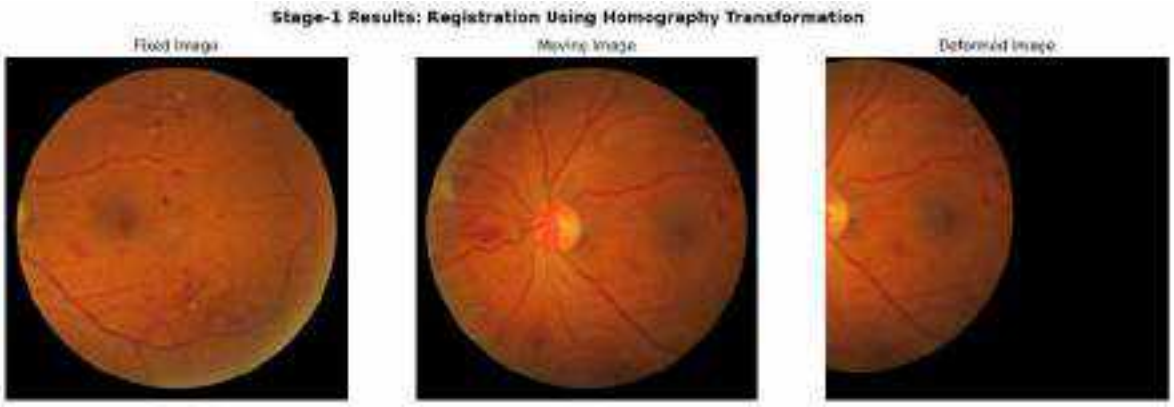
Note: 499 point correspondences were identified by the model for stage-2



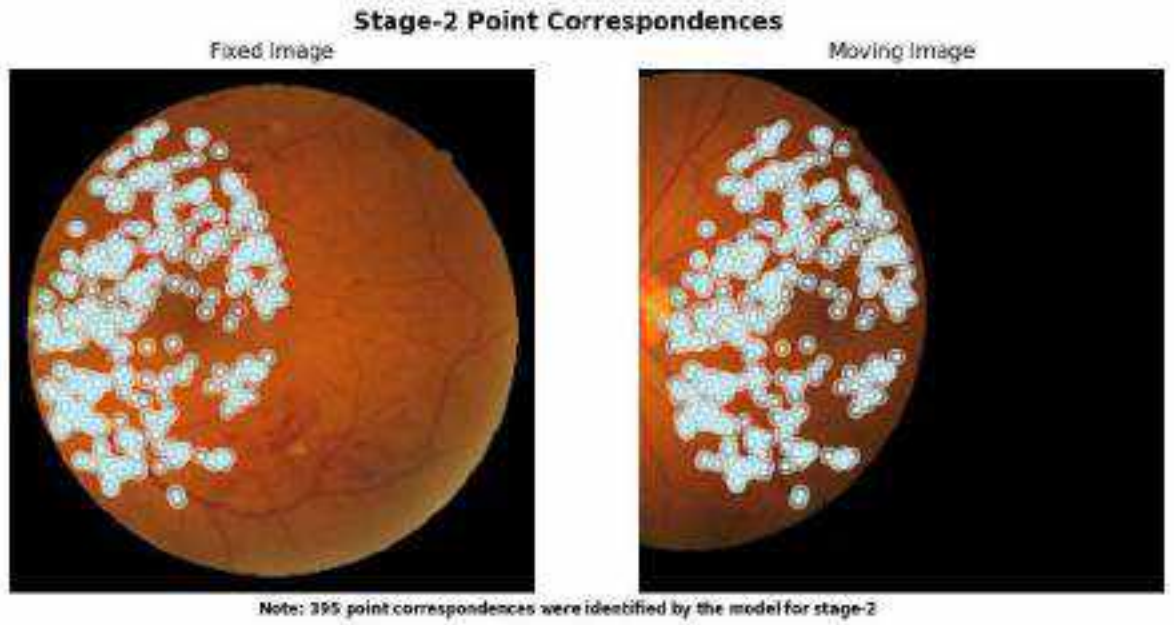

```
Mean Landmark Error for Case 20 Before Registration is 849.1492686429089 pixels
Mean Landmark Error for Case 20 After Registration is 2.2694640596393145 pixels
Case 21
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P22_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P22_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



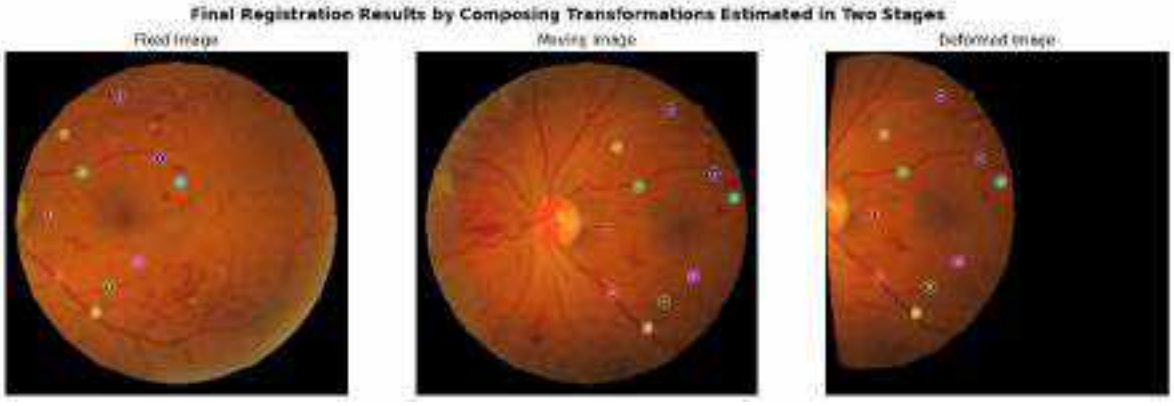

```
Homography Matrix:
[[ 8.93523732e-01 -1.81793499e-03 -3.38309785e+02]
 [-5.72700066e-02  9.20422599e-01  4.82628296e+00]
 [-1.02845836e-04 -8.29709335e-06  1.00000000e+00]]
```



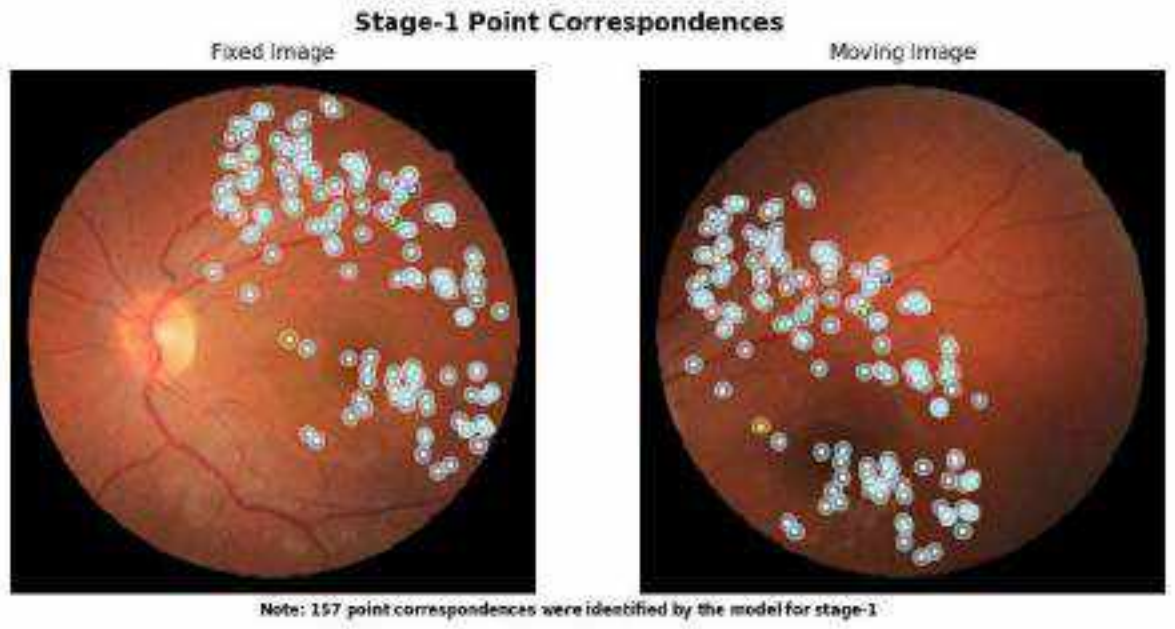

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



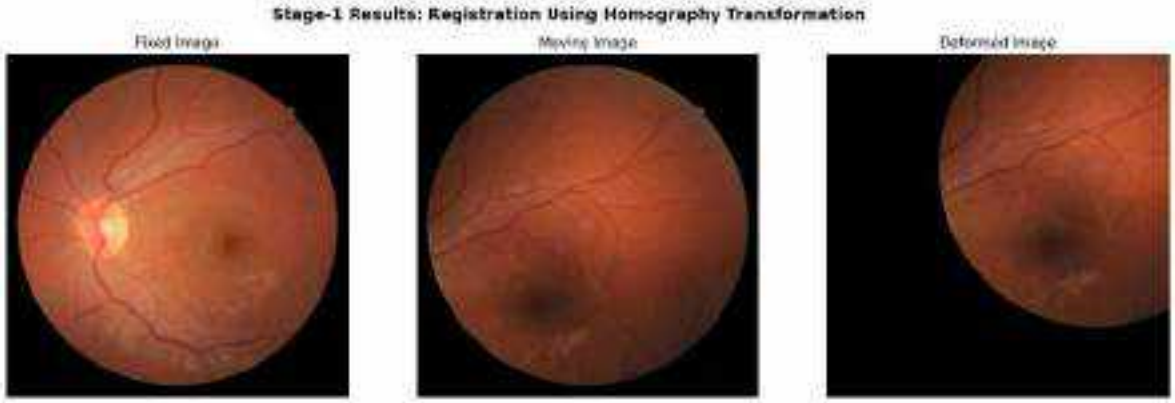
Note: 395 point correspondences were identified by the model for stage-2



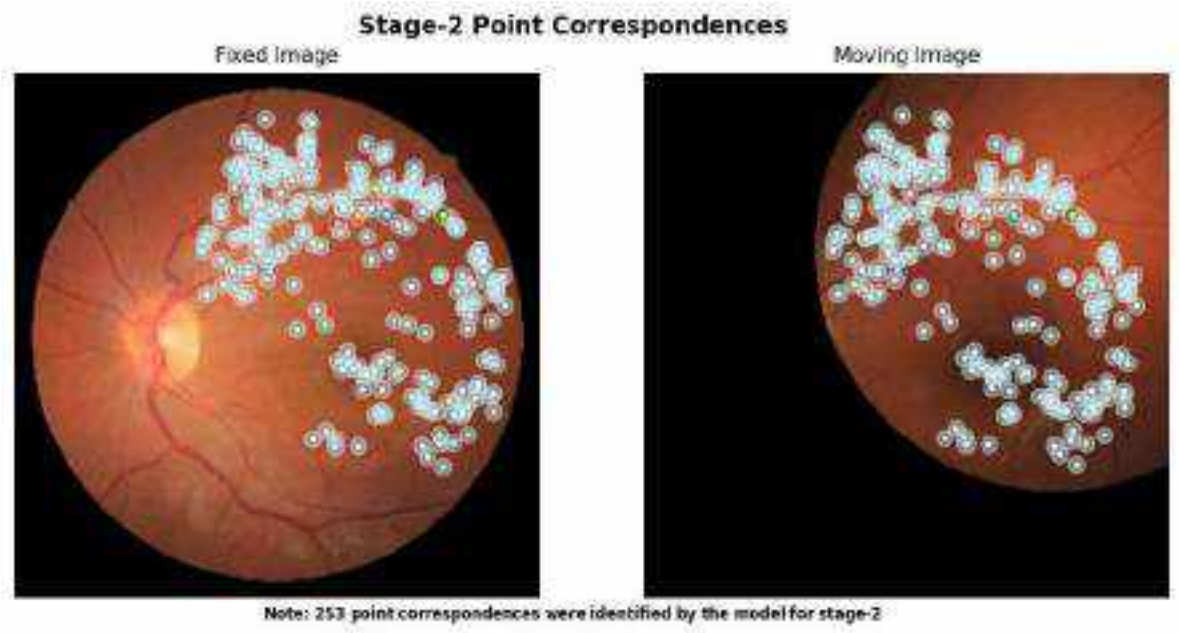
```
Mean Landmark Error for Case 21 Before Registration is 1231.616889161671 pixels
Mean Landmark Error for Case 21 After Registration is 2.4325843533411743 pixels
Case 22
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P23_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P23_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



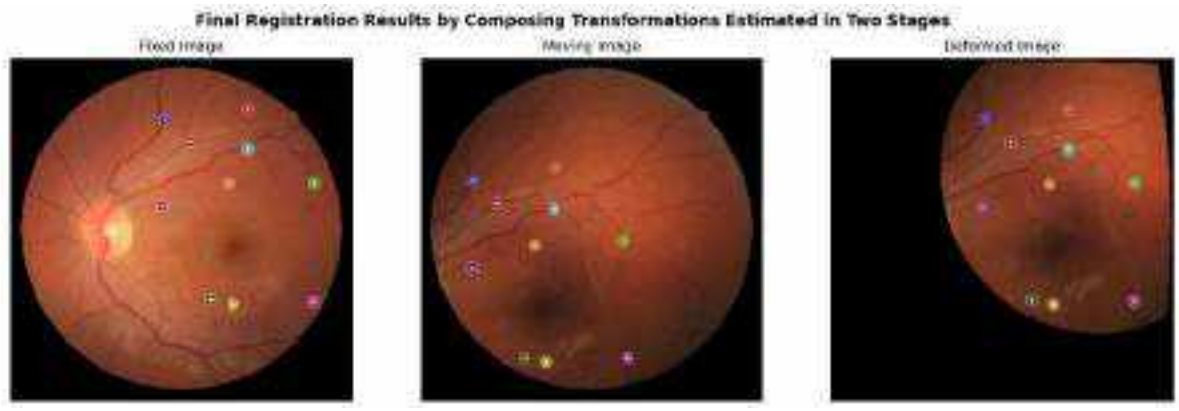

```
Homography Matrix:
[[ 1.04611831e+00 -4.32639755e-02  2.84251038e+02]
 [ 4.97506728e-02  9.93088982e-01 -1.70806205e+02]
 [ 6.700560520-05 -3.09455502e-05  1.00000000e+00]]
```



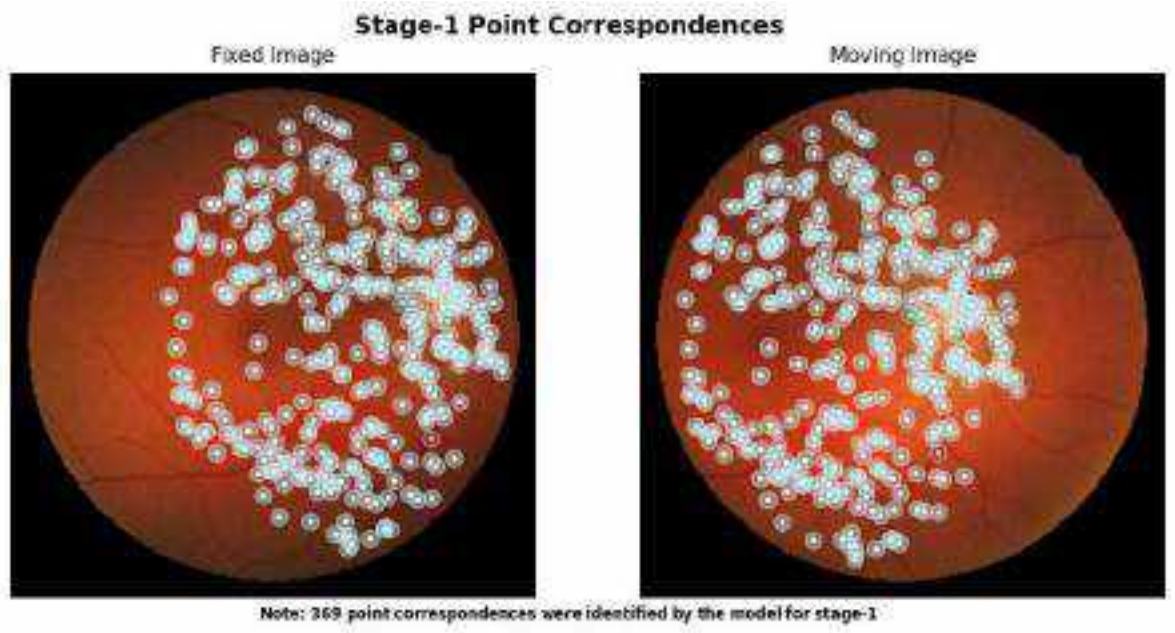

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 253 point correspondences were identified by the model for stage-2



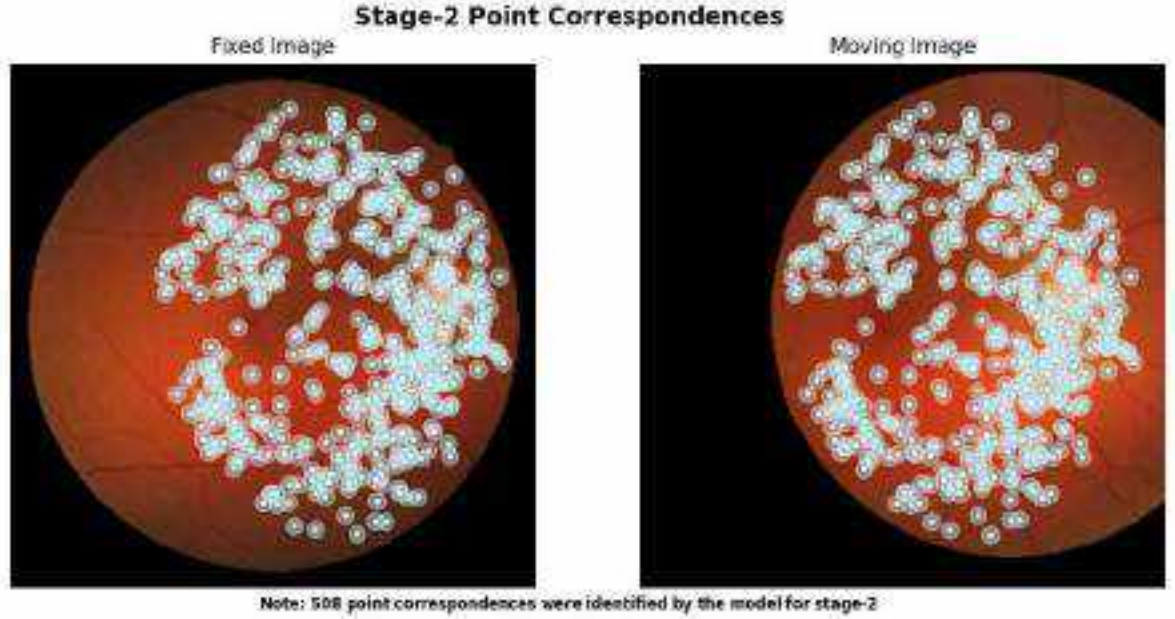

```
Mean Landmark Error for Case 22 Before Registration is 995.9820480773485 pixels
Mean Landmark Error for Case 22 After Registration is 3.1484343523754 77 pixels
Case 23
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P24_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P24_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



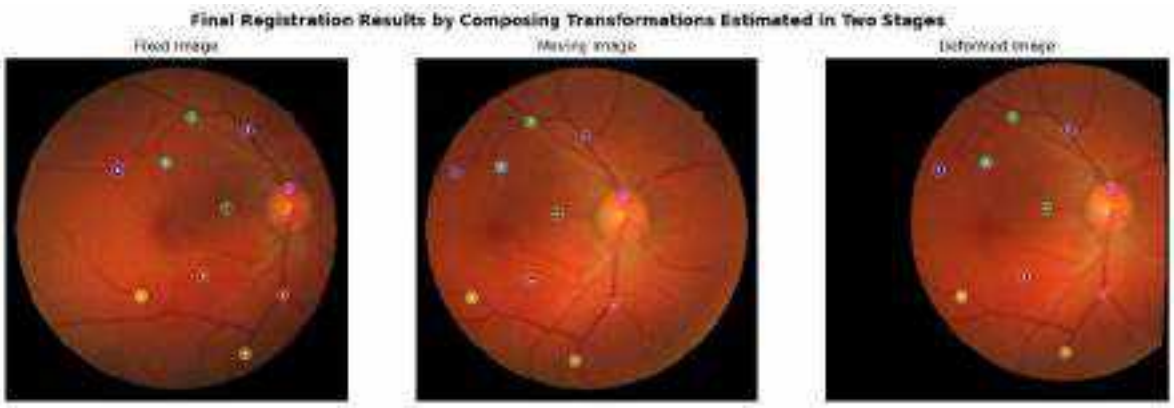
Note: 369 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.09643006e+00  5.15609794e-02  1.75034652e+02]
 [ 1.96586209e-03  1.04021962e+00 -1.58789321e+01]
 [ 1.05976168e-04  6.88793525e-06  1.00000000e+00]]
```

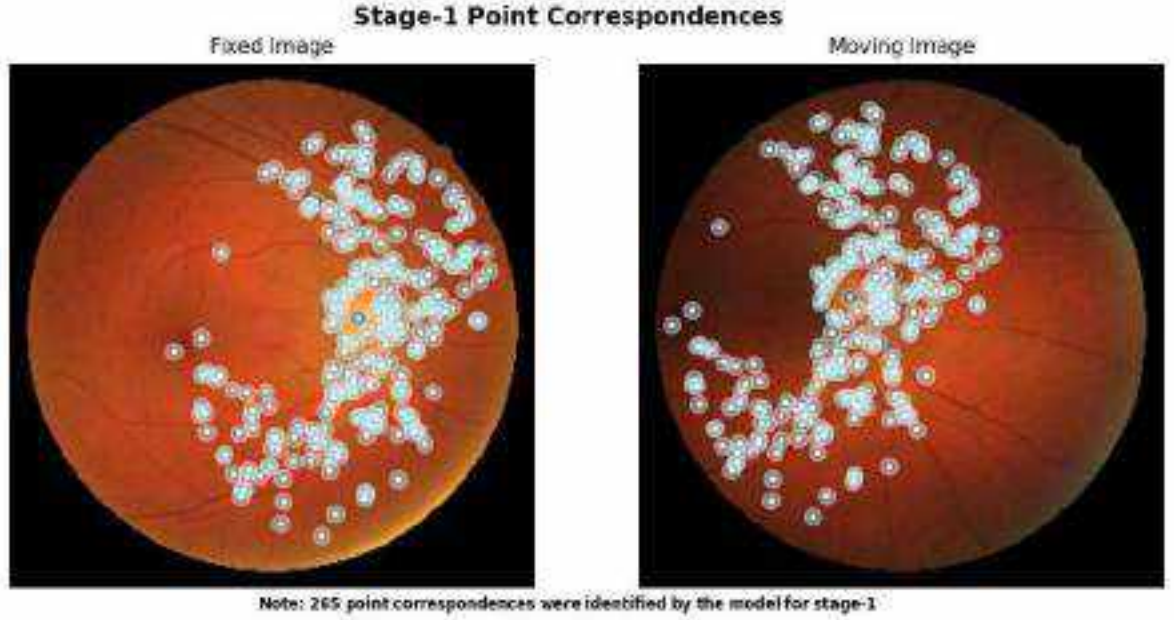

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



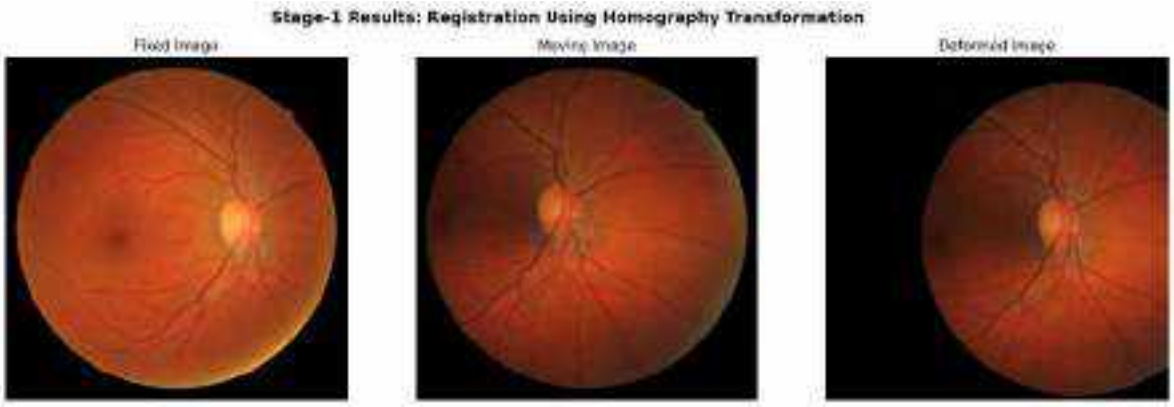
Note: 508 point correspondences were identified by the model for stage-2



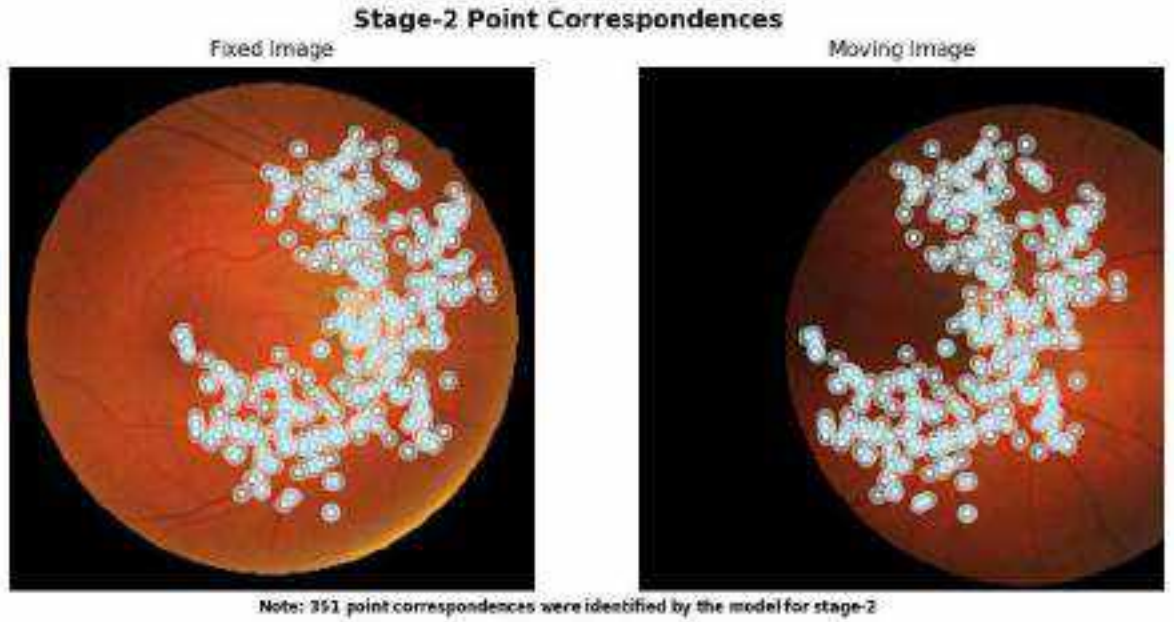

```
Mean Landmark Error for Case 23 Before Registration is 654.6451606282154 pixels
Mean Landmark Error for Case 23 After Registration is 2.4996476664825407 pixels
Case 24
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P25_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P25_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



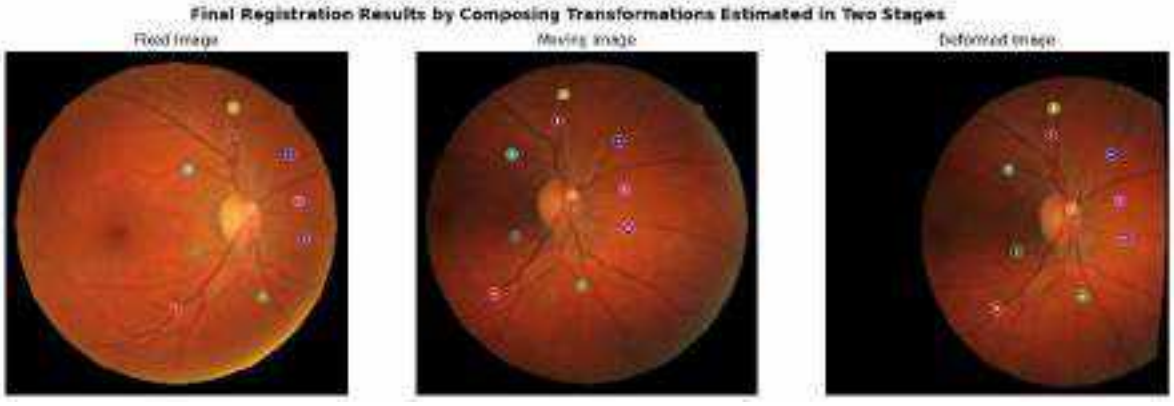

```
Homography Matrix:
[[1.14828879e+00 7.32859688e-02 1.95300537e+02]
 [2.93430137e-02 1.07046045e+00 2.78046482e+01]
 [1.53203515e-04 2.92033556e-05 1.00000000e+00]]
```



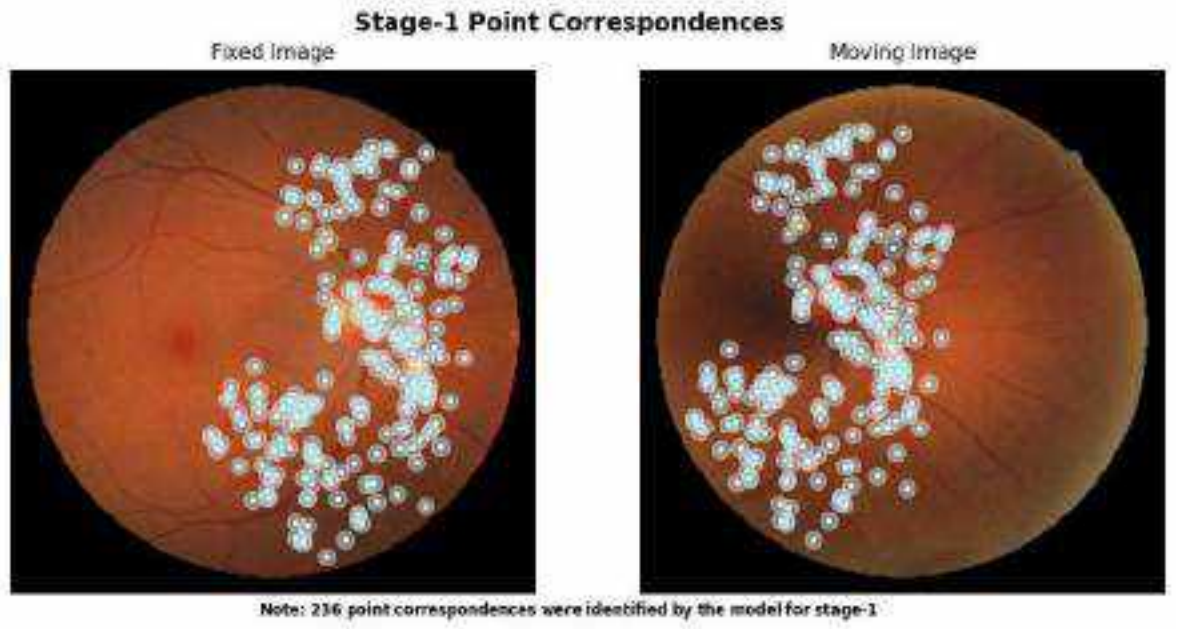

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



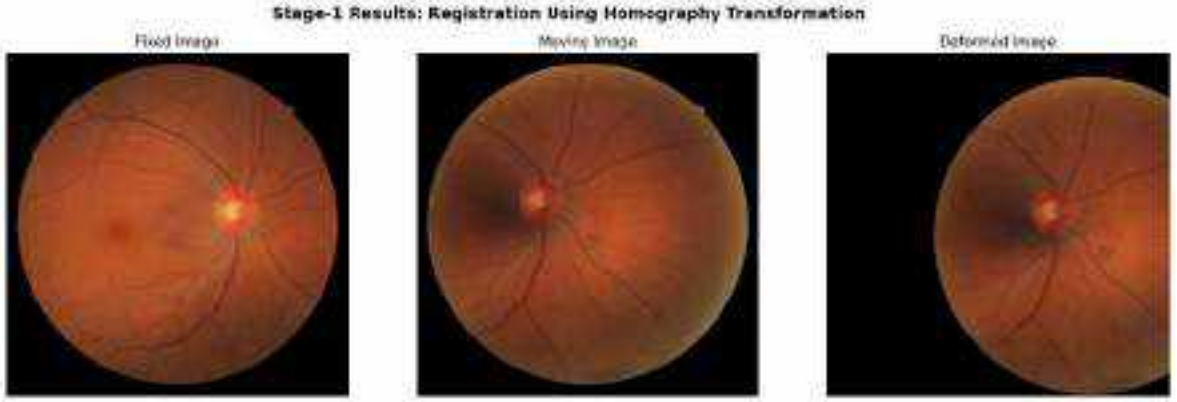
Note: 351 point correspondences were identified by the model for stage-2



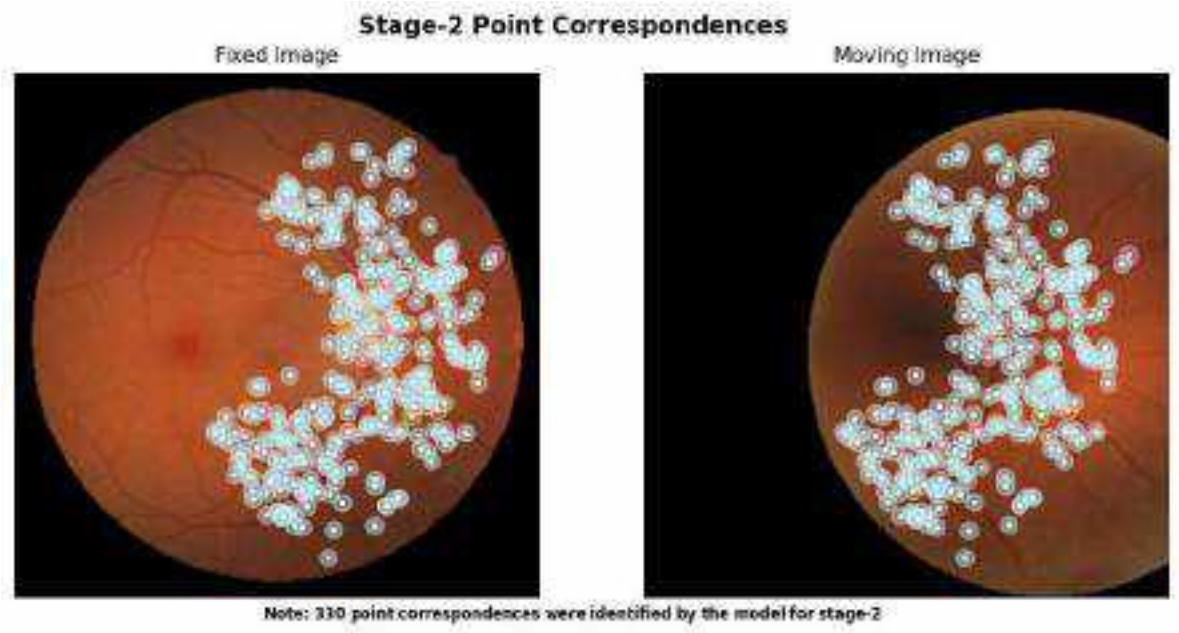
```
Mean Landmark Error for Case 24 Before Registration is 747.7321675337728 pixels
Mean Landmark Error for Case 24 After Registration is 3.046520686429399 pixels
Case 25
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P26_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P26_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

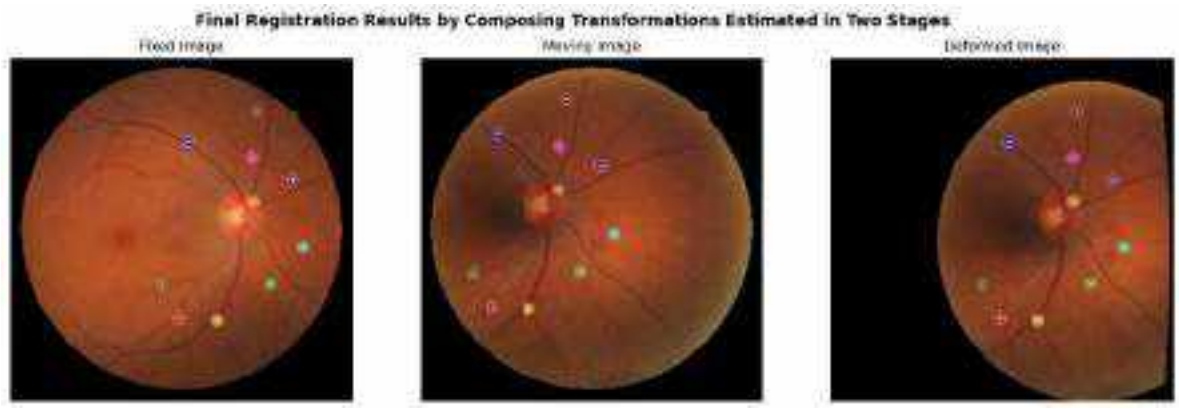



```
Homography Matrix:
[[ 1.14639312e+00 -1.12037368e-02  2.61033323e+02]
 [ 8.88533169e-02  1.06266112e+00 -1.34505577e+00]
 [ 1.36662533e-04  1.47199596e-05  1.00000000e+00]]
```

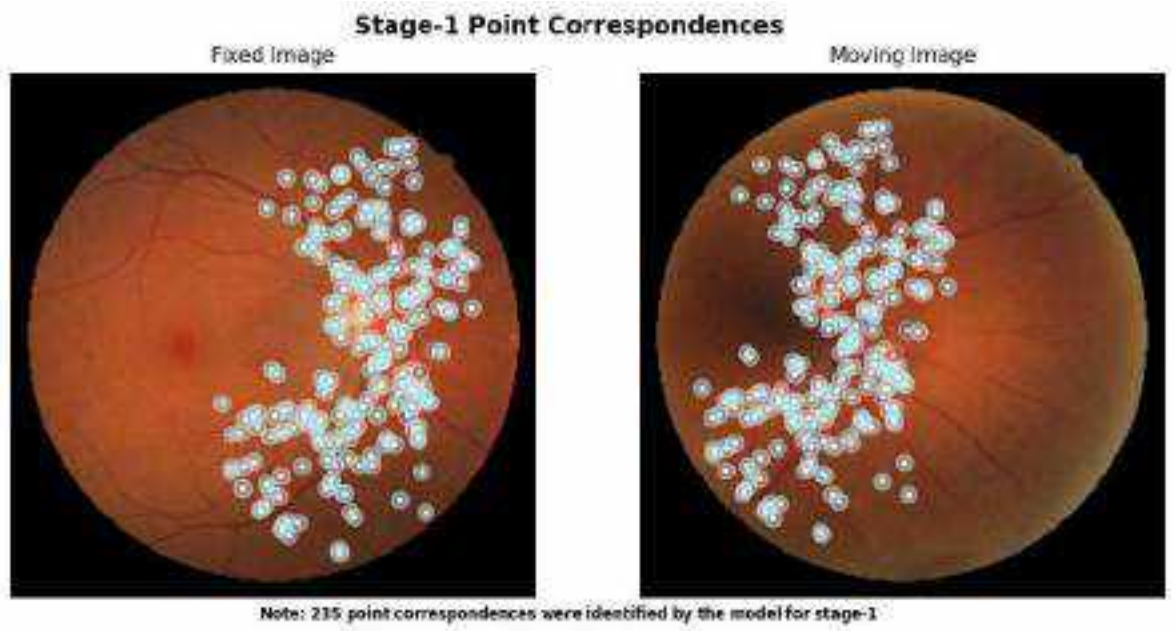



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





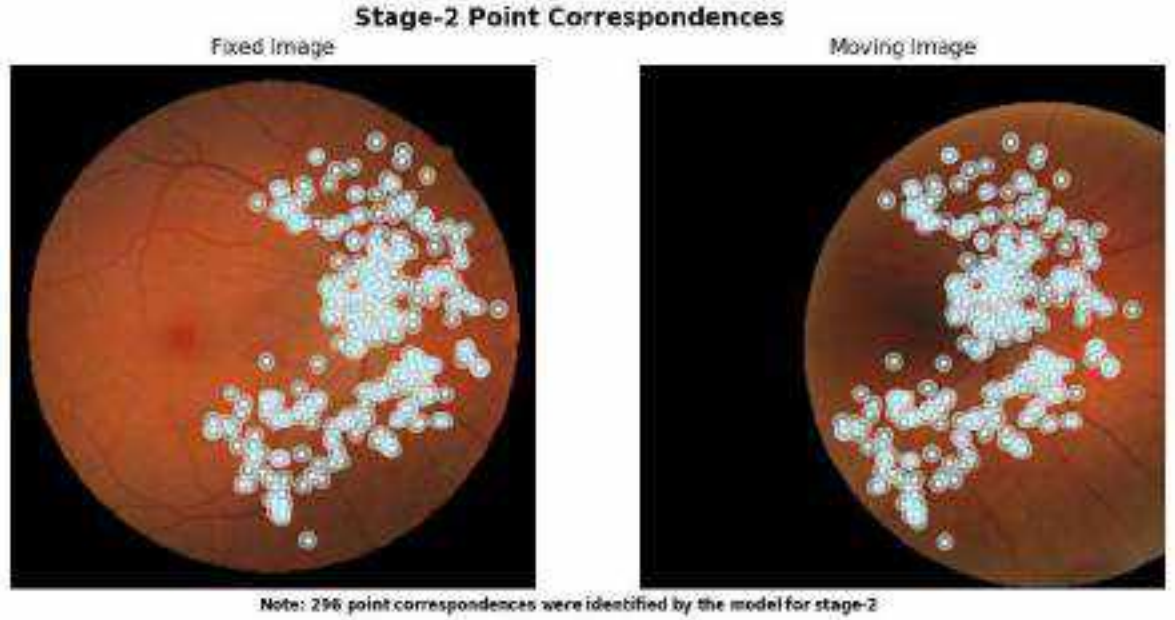

```
Mean Landmark Error for Case 25 Before Registration is 862.6337096837835 pixels
Mean Landmark Error for Case 25 After Registration is 2.7122088325099623 pixels
Case 26
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P27_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P27_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



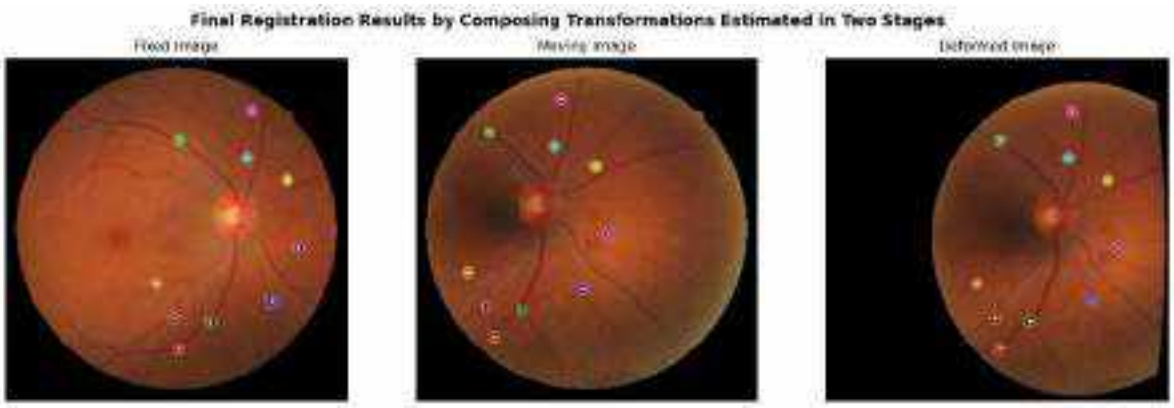
Note: 235 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.15724361e+00 -1.35656177e-02  2.61194455e+02]
 [ 9.61070600e-02  1.05487108e+00 -3.29390009e+00]
 [ 1.52241448e-04  1.02922006e-05  1.00000000e+00]]
```

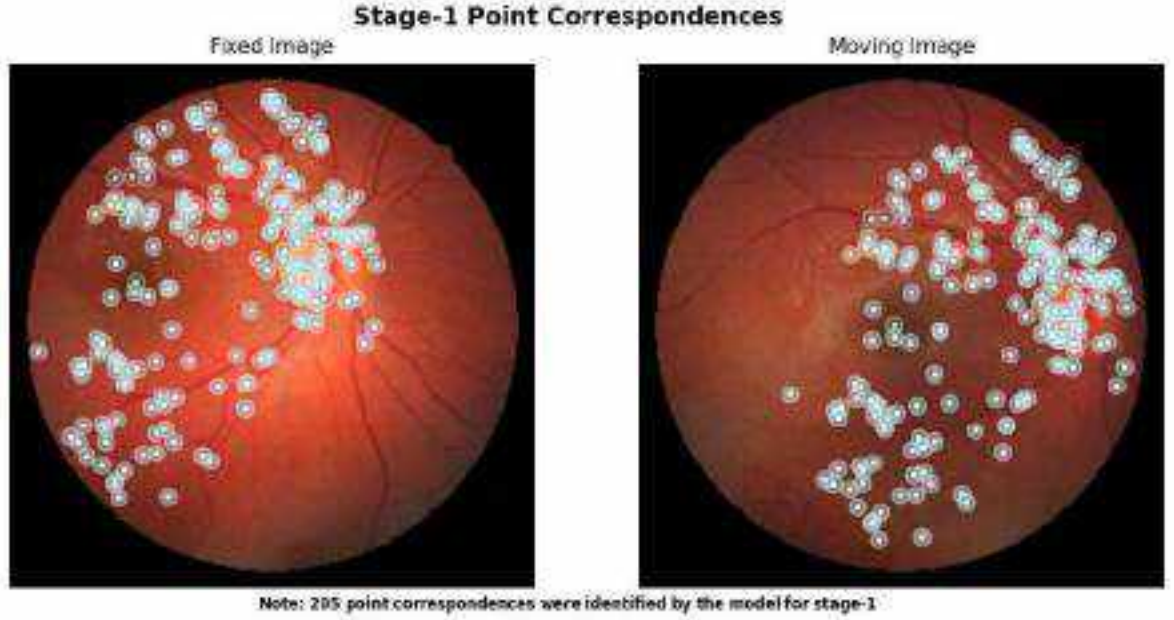

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 296 point correspondences were identified by the model for stage-2

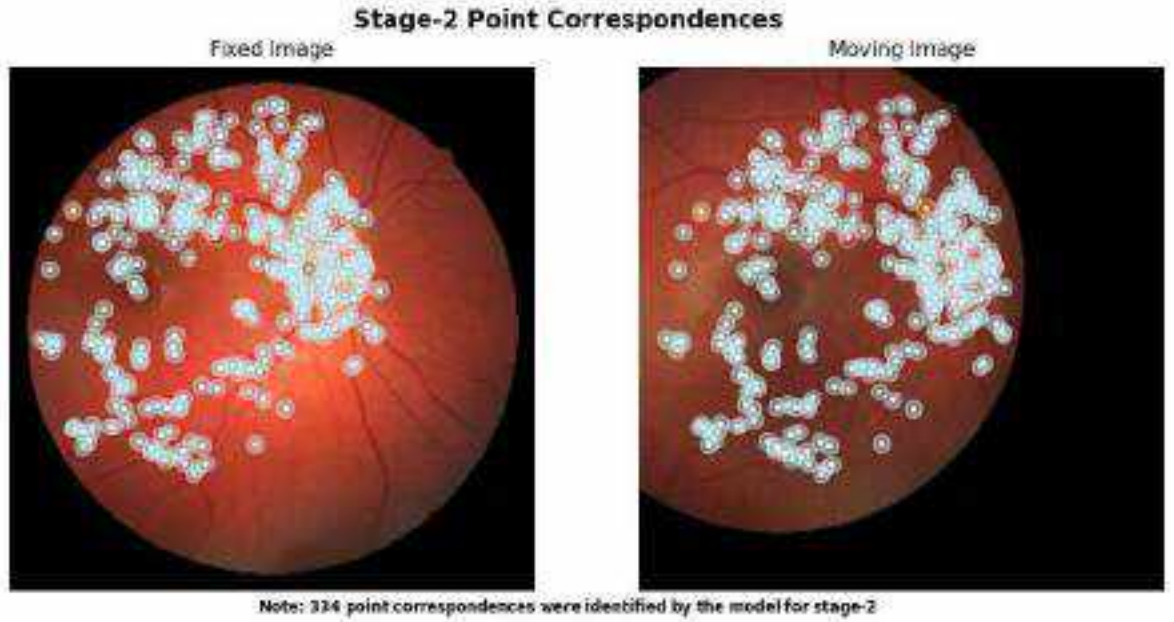



```
Mean Landmark Error for Case 26 Before Registration is 853.5424767473085 pixels
Mean Landmark Error for Case 26 After Registration is 4.8937649930923765 pixels
Case 27
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P28_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P28_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

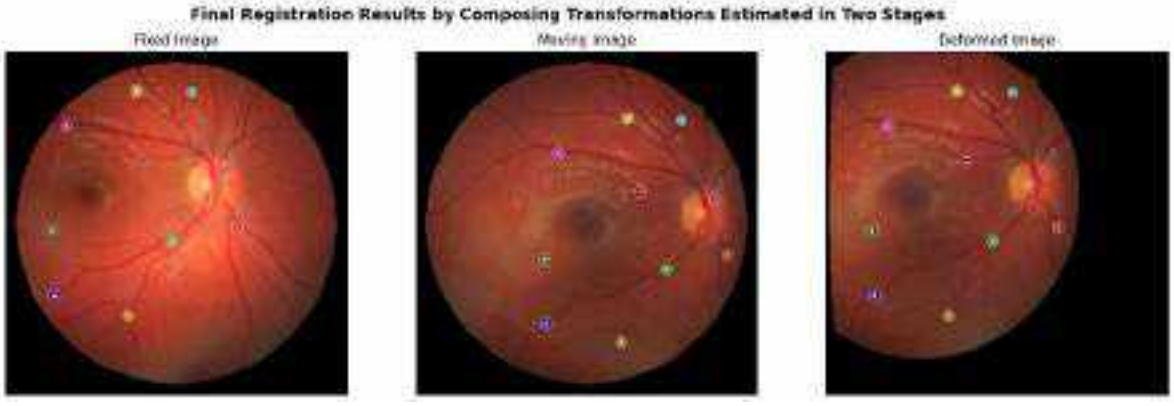



```
Homography Matrix:
[[ 9.21009039e-01 -2.16126738e-02 -1.89581560e+02]
 [-3.34589657e-02  9.37053265e-01 -5.00428654e+01]
 [-8.15581063e-05 -2.53030531e-05  1.00000000e+00]]
```

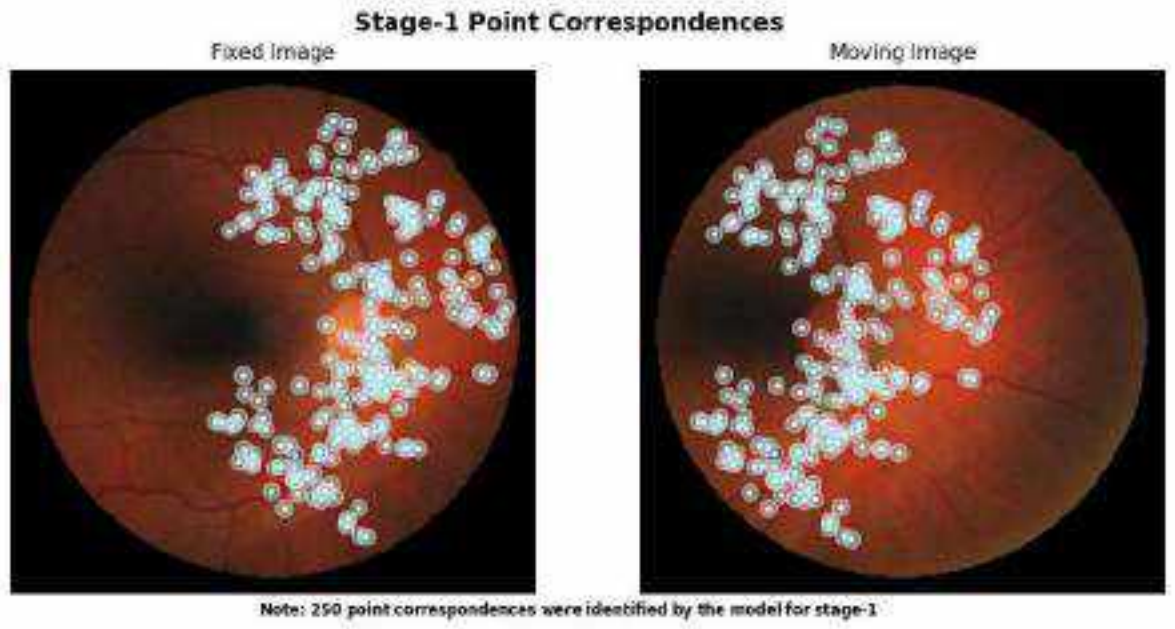

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



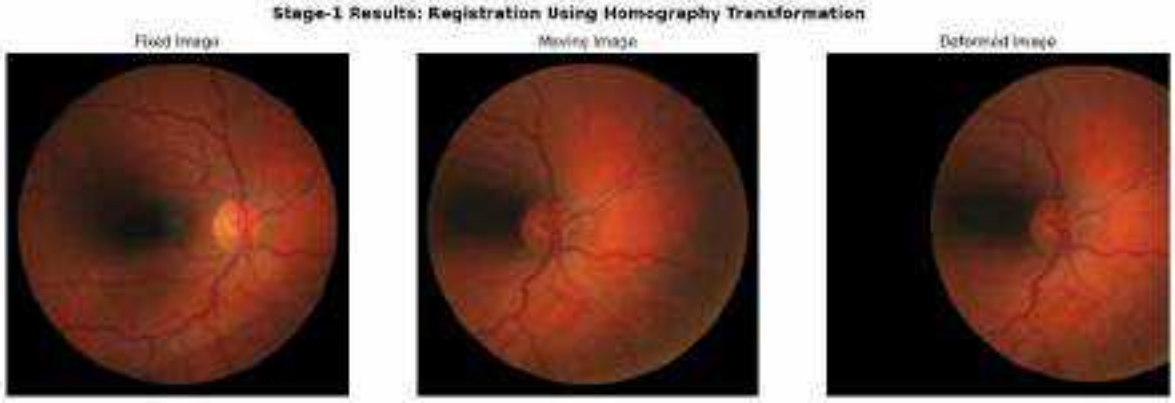
Note: 314 point correspondences were identified by the model for stage-2



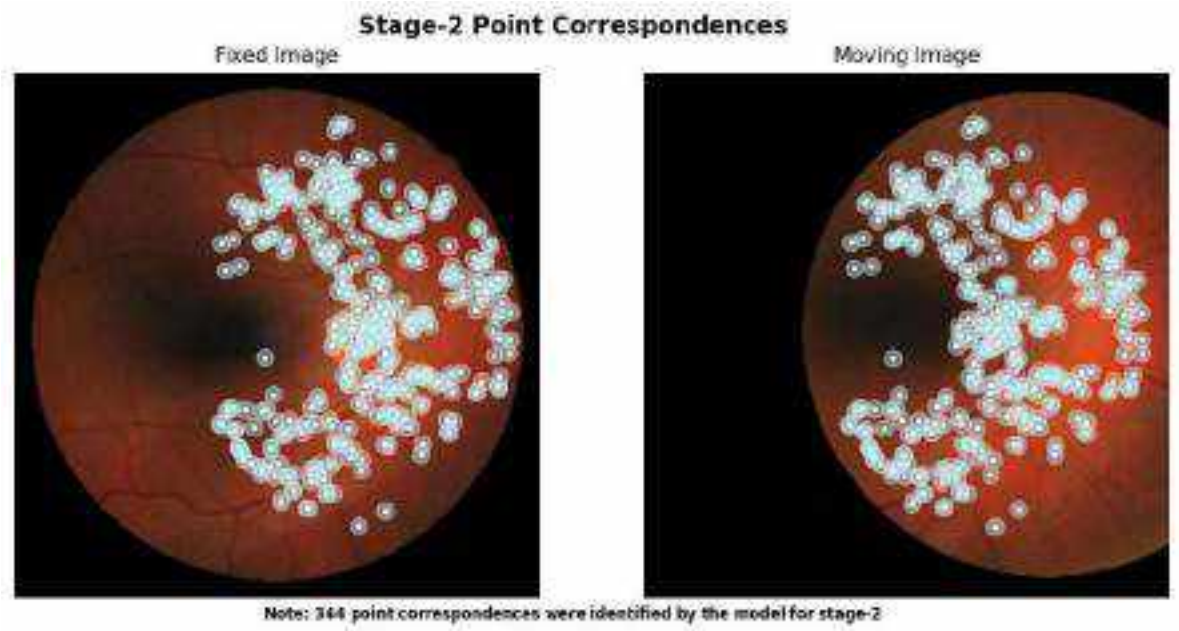
```
Mean Landmark Error for Case 27 Before Registration is 736.4352092342231 pixels
Mean Landmark Error for Case 27 After Registration is 3.0642033672503723 pixels
Case 28
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P29_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P29_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



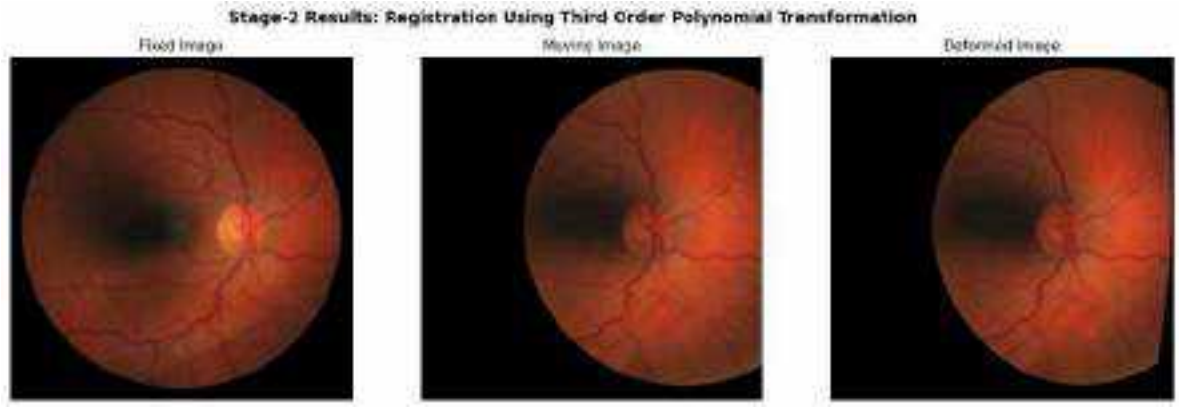

```
Homography Matrix:
[[ 1.11217355e+00  1.08321804e-02  2.38413135e+02]
 [ 4.90857335e-02  1.02266860e+00 -1.62783395e+01]
 [ 1.20632733e-04 -1.88786023e-05  1.00000000e+00]]
```

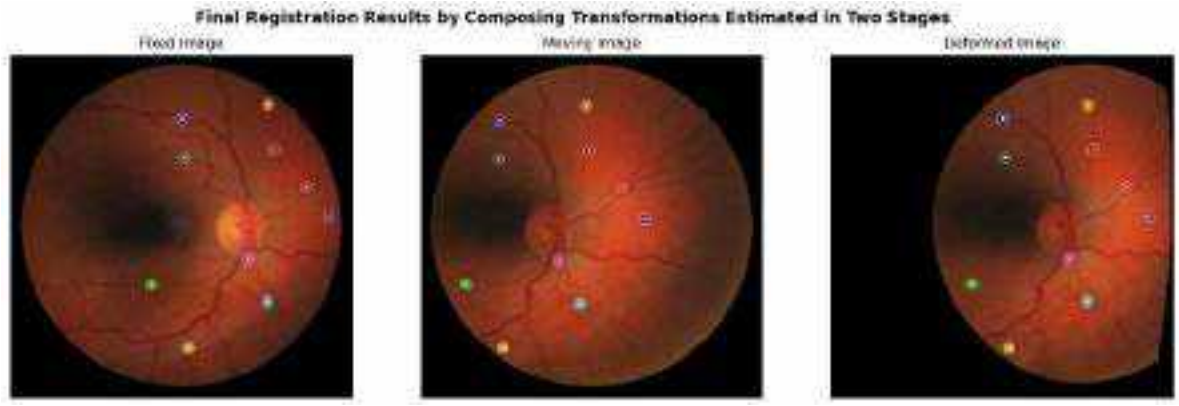



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

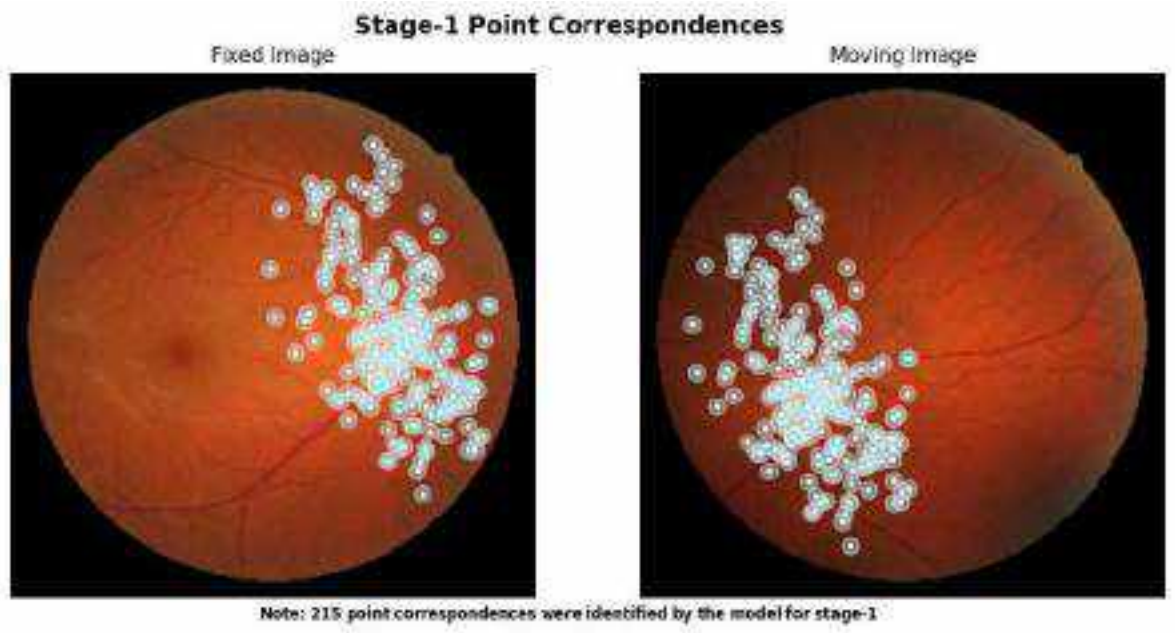


Note: 344 point correspondences were identified by the model for stage-2





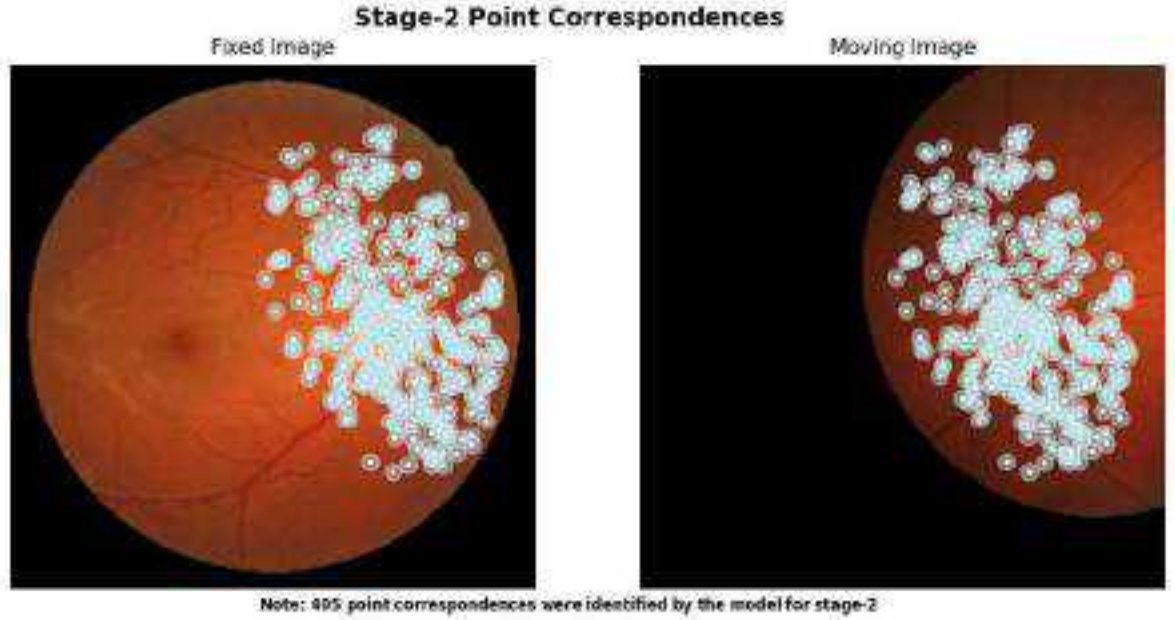

```
Mean Landmark Error for Case 28 Before Registration is 811.4847761859805 pixels
Mean Landmark Error for Case 28 After Registration is 3.042775486290286 pixels
Case 29
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P30_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P30_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



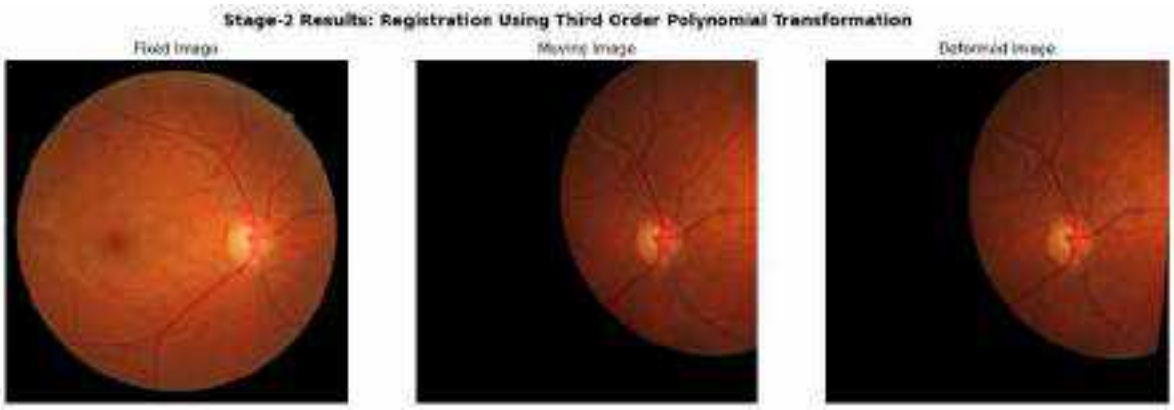
Note: 215 point correspondences were identified by the model for stage-1

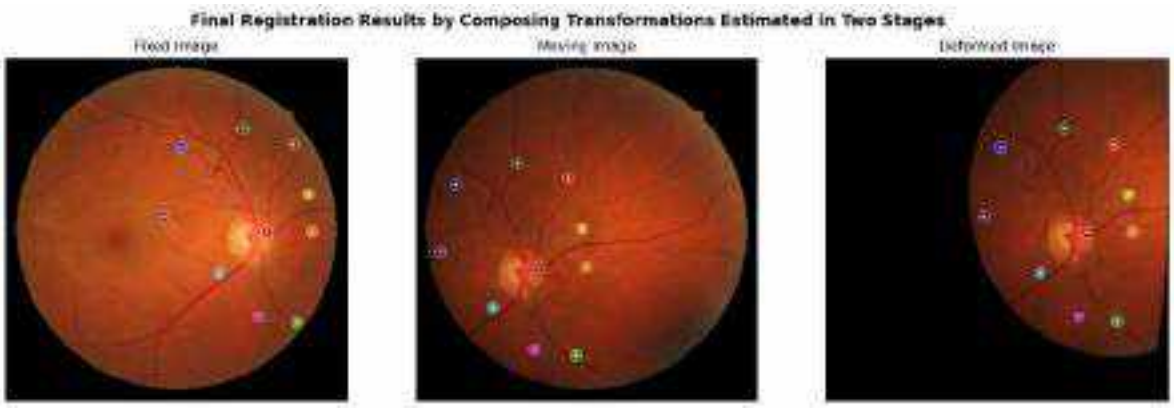
```
Homography Matrix:
[[ 1.17024599e+00 -3.99722231e-02  3.66840743e+02]
 [ 1.10715681e-01  1.01688416e+00 -1.20922512e+02]
 [ 2.20052237e-04 -5.46884622e-05  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

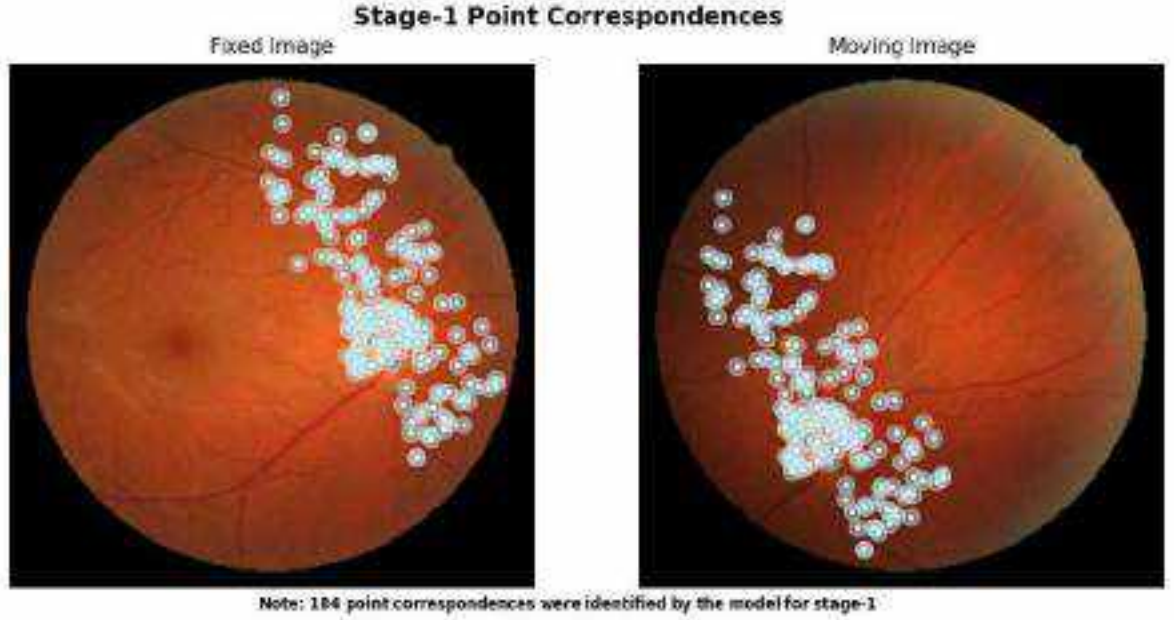


Note: 495 point correspondences were identified by the model for stage-2

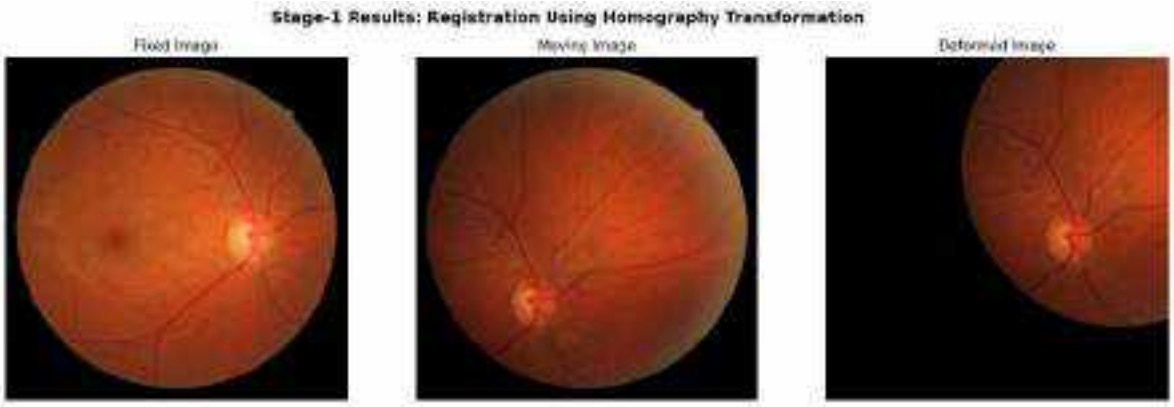




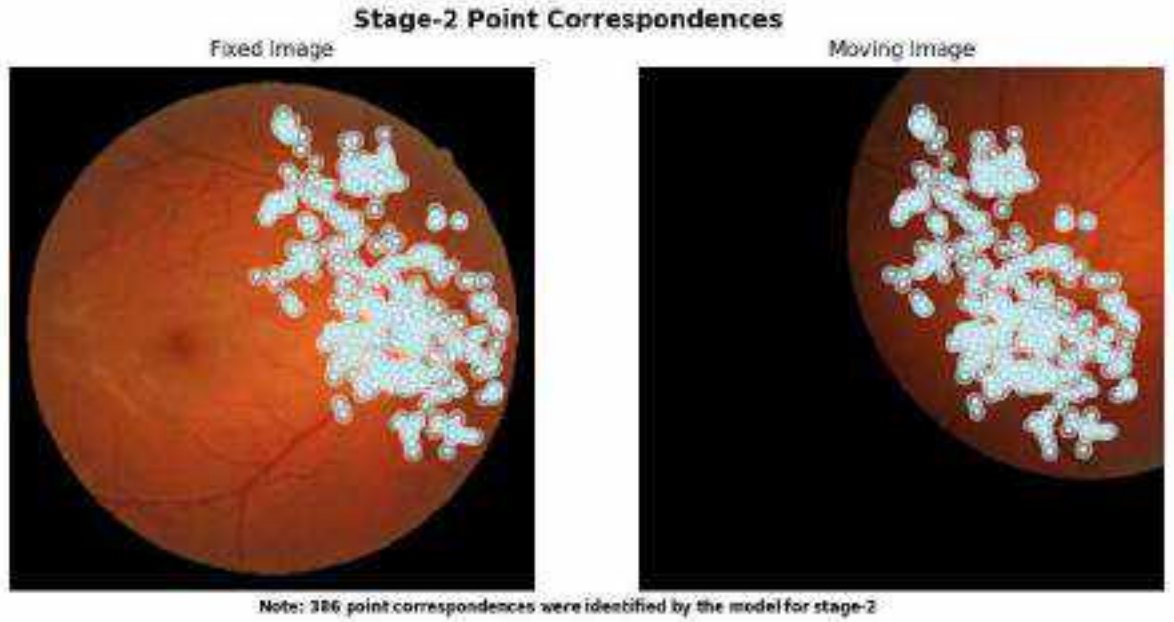

```
Mean Landmark Error for Case 29 Before Registration is 1189.40284017284447 pixels
Mean Landmark Error for Case 29 After Registration is 4.011433586186048 pixels
Case 30
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P31_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P31_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



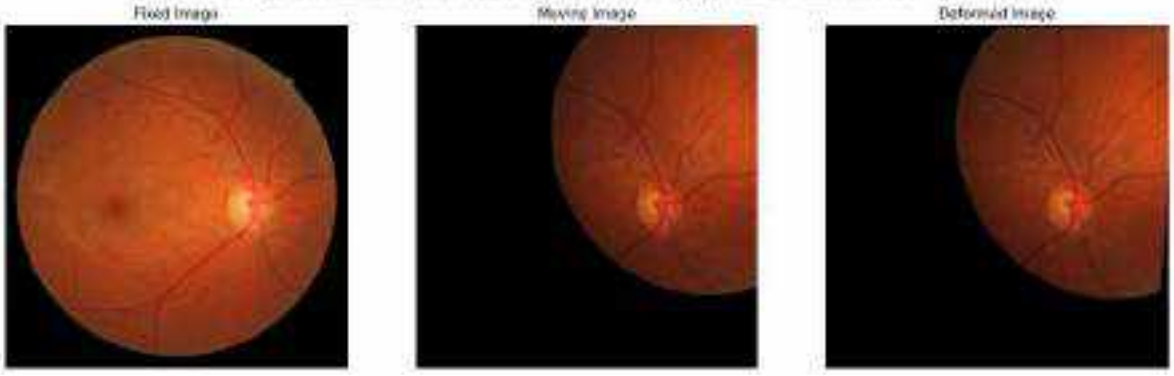

```
Homography Matrix:
[[ 1.06770780e+00 -6.97922589e-02  3.51950395e+02]
 [ 9.78430659e-02  9.63660566e-01 -1.86515487e+02]
 [ 1.40251136e-04 -8.56231505e-05  1.00000000e+00]]
```

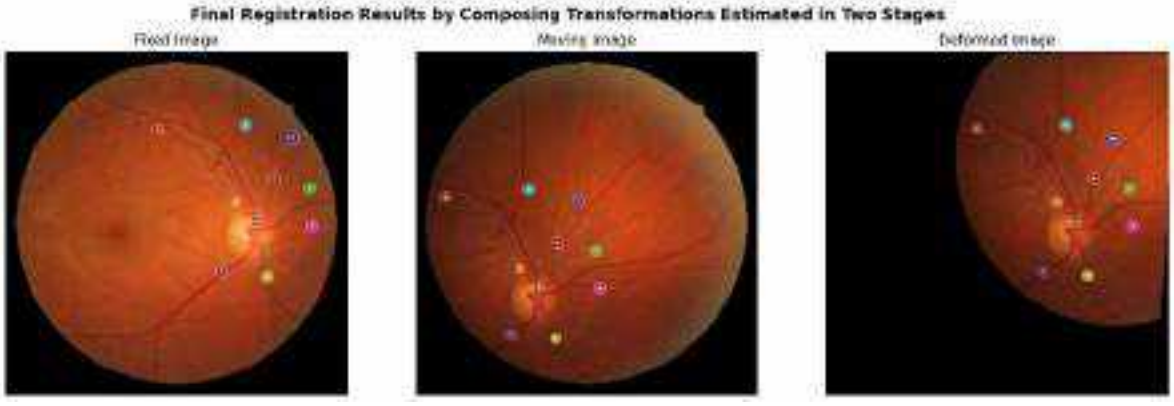



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

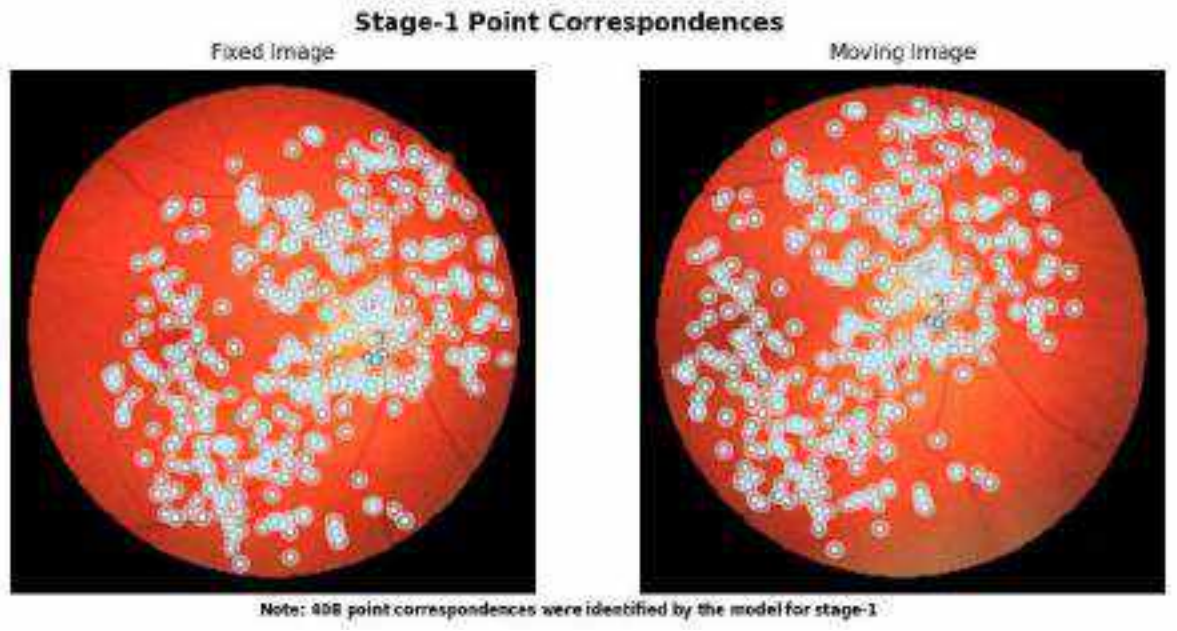


Note: 386 point correspondences were identified by the model for stage-2

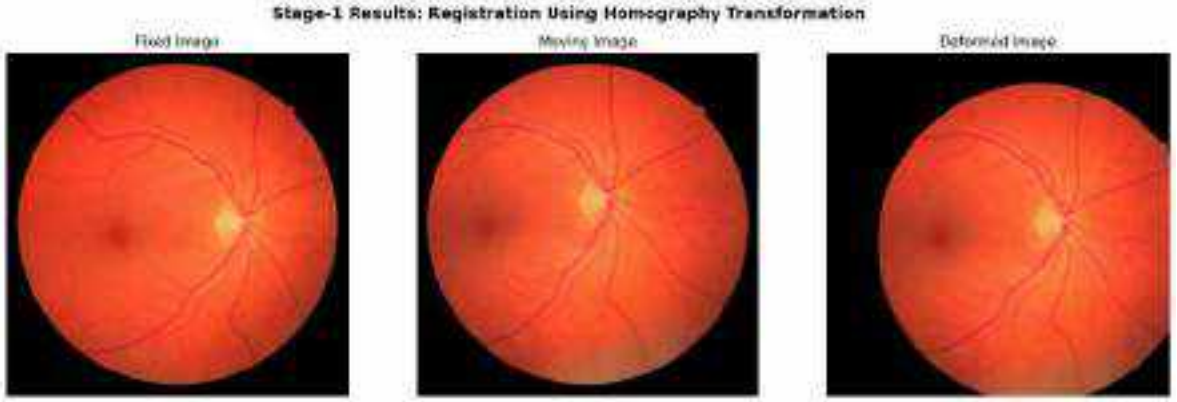




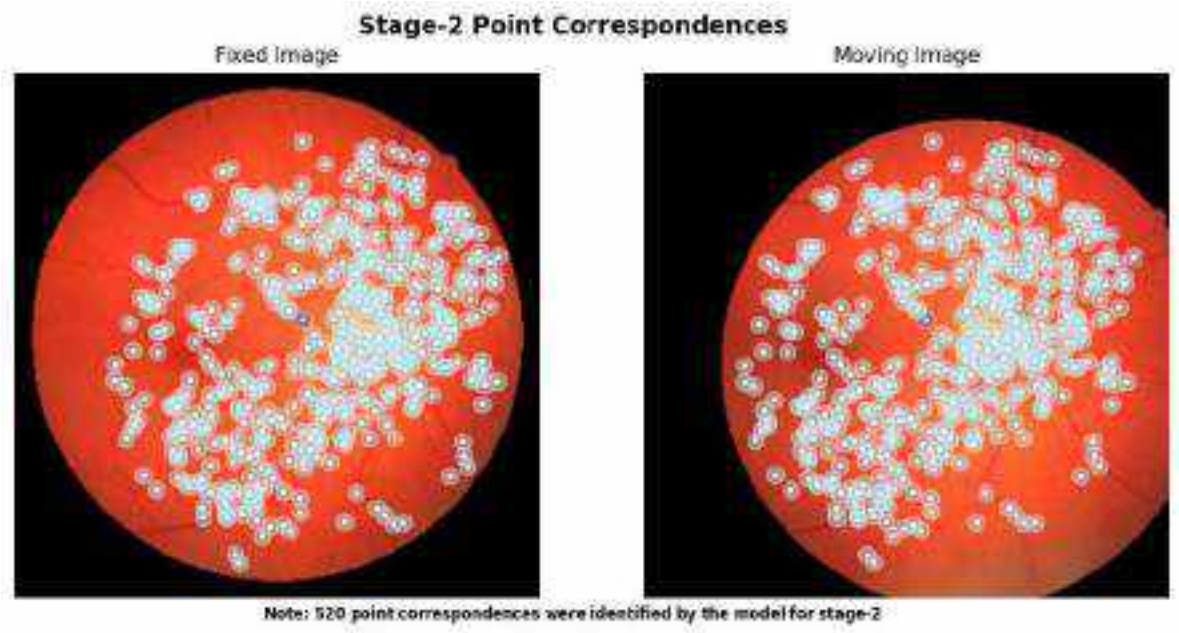
```
Mean Landmark Error for Case 30 Before Registration is 1191.213230739895 pixels
Mean Landmark Error for Case 30 After Registration is 5.003432585956O165 pixels
Case 31
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P32_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P32_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



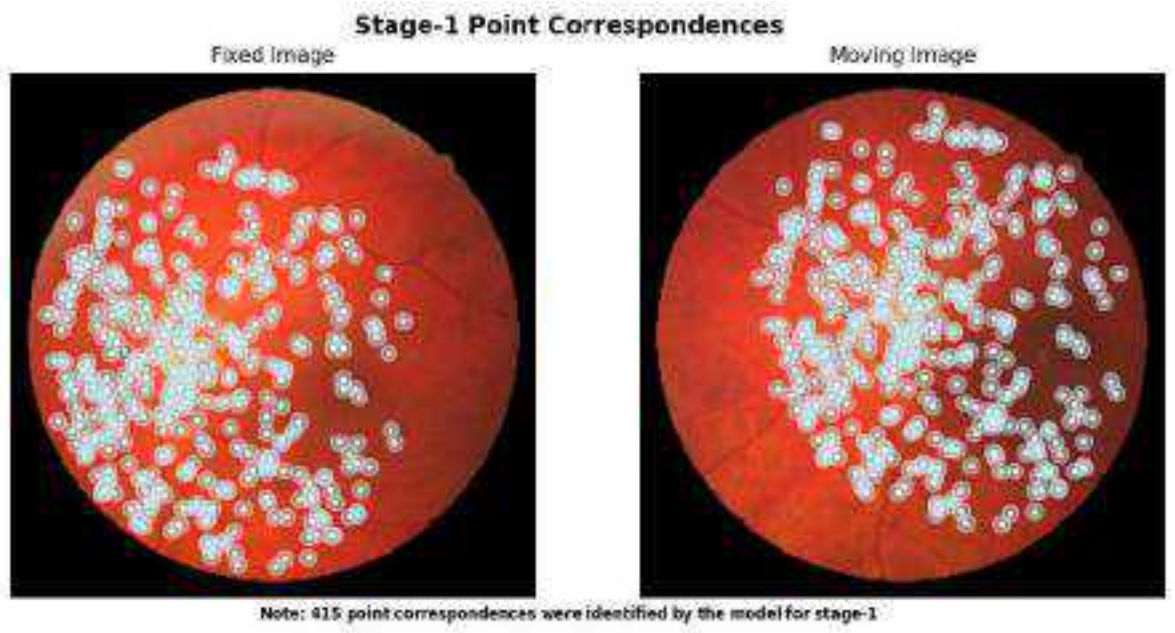

```
Homography Matrix:
[[ 1.06104588e+00 -1.07817653e-01  1.58300063e+02]
 [ 1.58651284e-01  1.03738275e+00 -1.04810265e+01]
 [ 6.56910350e-05  1.98690750e-05  1.00000000e+00]]
```




```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



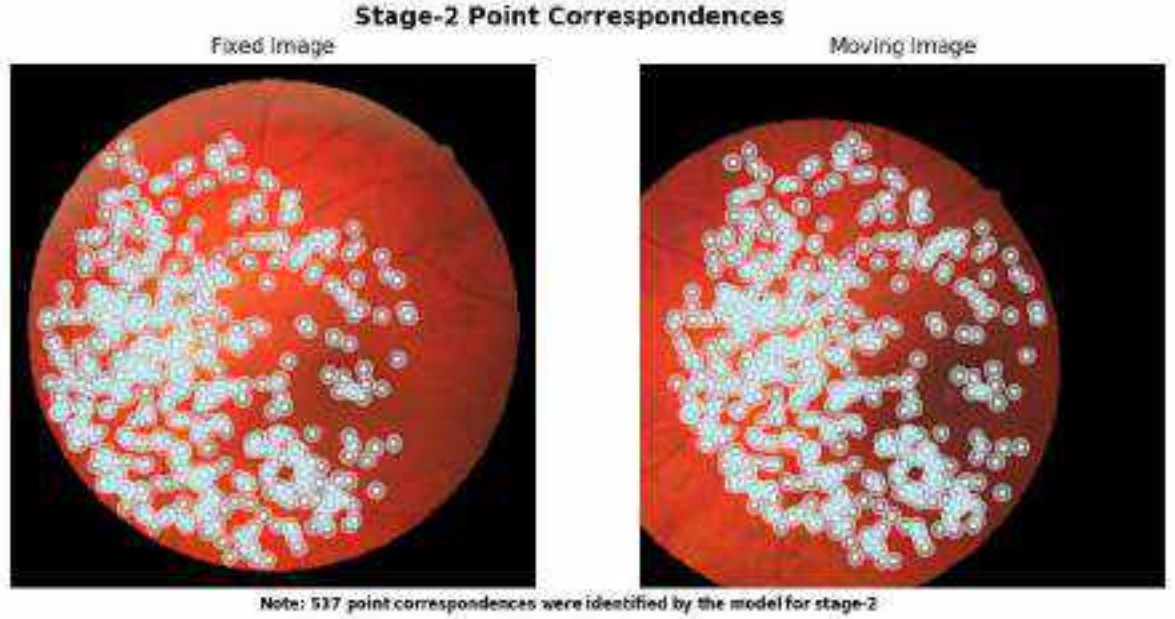

```
Mean Landmark Error for Case 31 Before Registration is 407.65168565965524 pixels
Mean Landmark Error for Case 31 After Registration is 4.387032606035843 pixels
Case 32
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P33_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P33_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



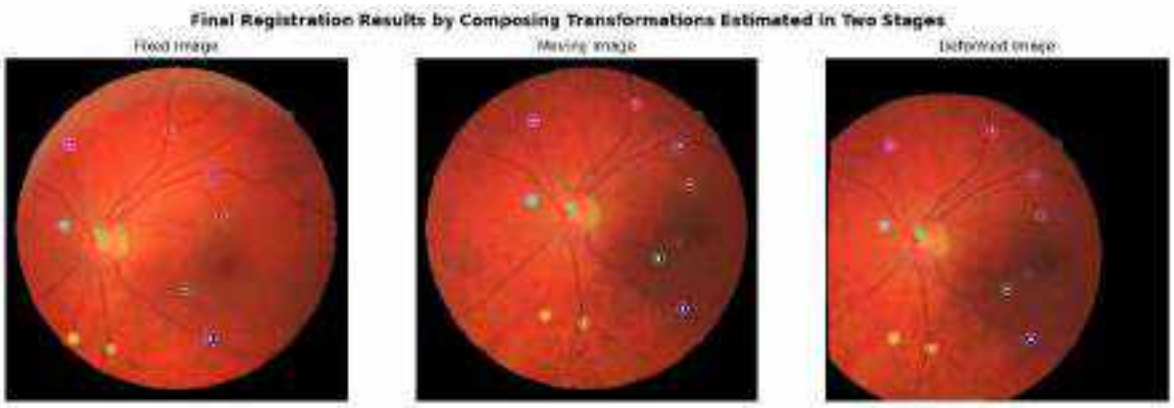
Note: 415 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.38808386e-01 -2.38253465e-02 -1.28721877e+02]
 [-1.11386213e-03  9.94808635e-01  6.54291067e+01]
 [-8.50180549e-05  4.45881554e-05  1.00000000e+00]]
```

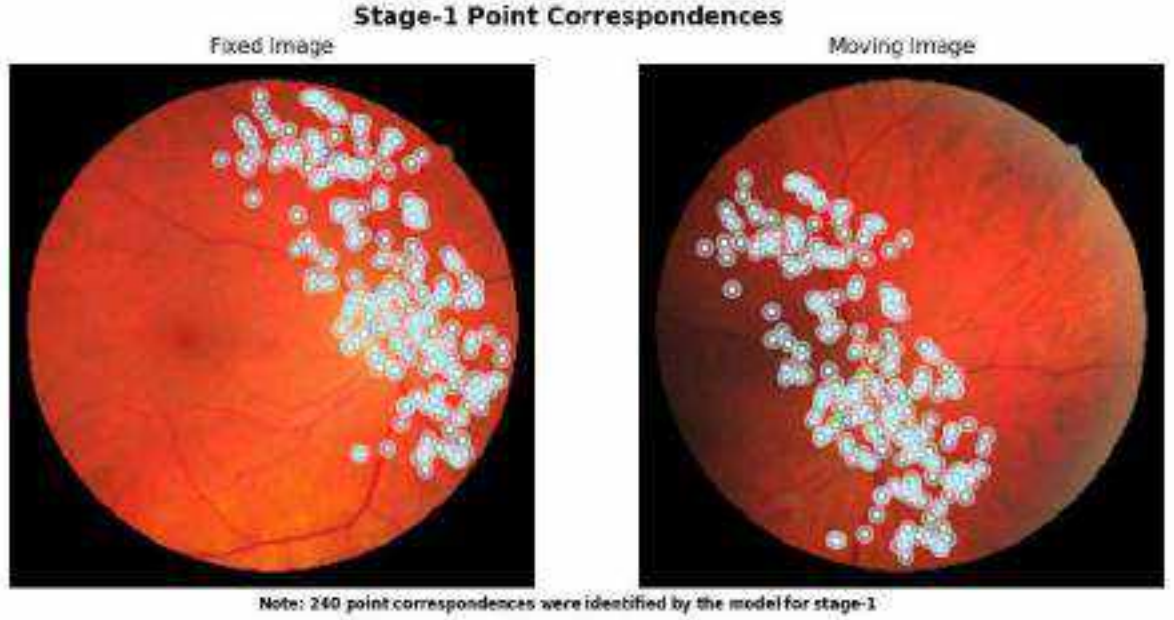

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



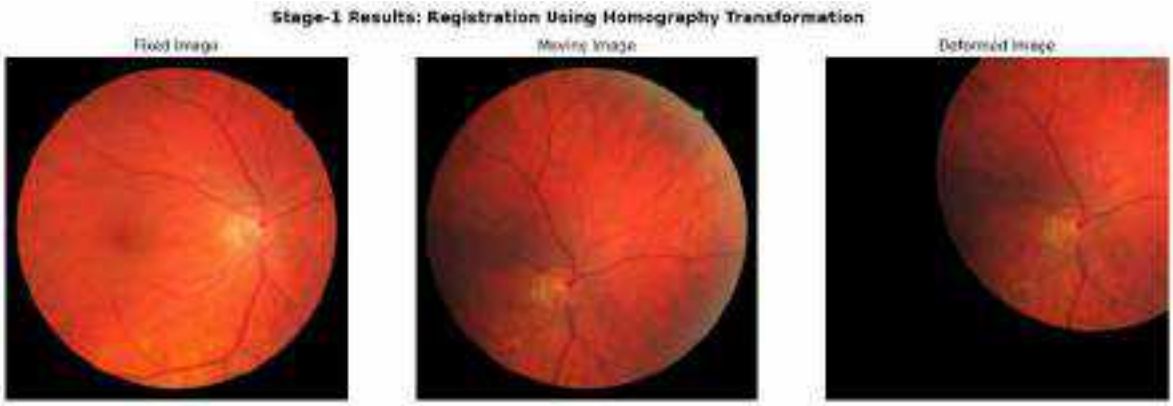
Note: 537 point correspondences were identified by the model for stage-2



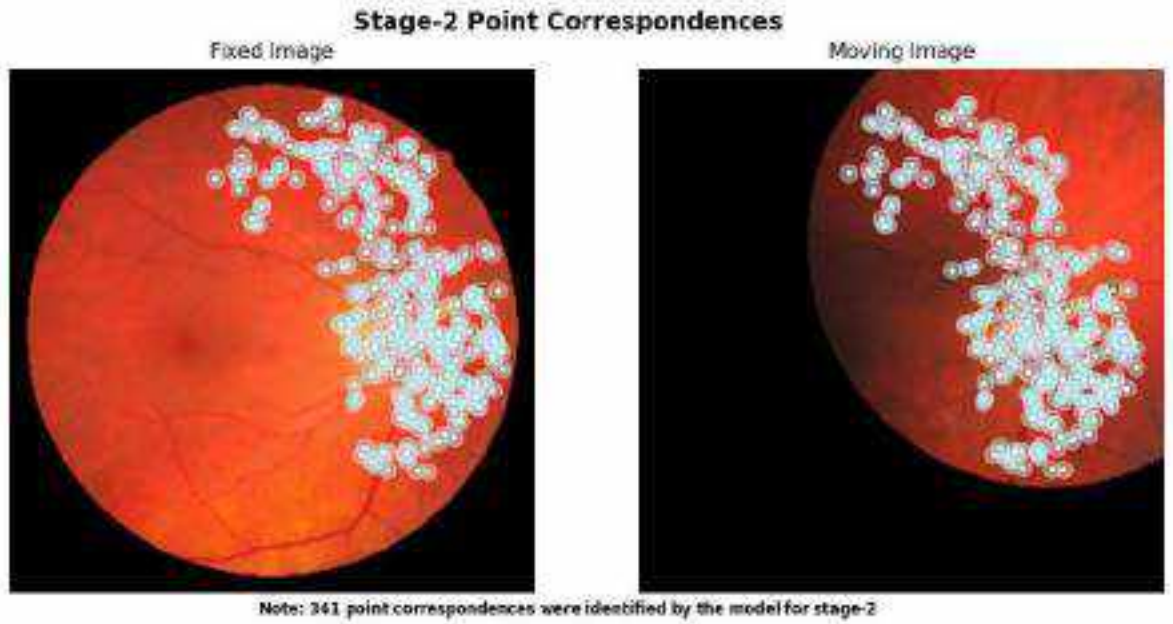

```
Mean Landmark Error for Case 32 Before Registration is 553.257106446283 pixels
Mean Landmark Error for Case 32 After Registration is 3.0170363624682635 pixels
Case 33
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P34_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P34_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



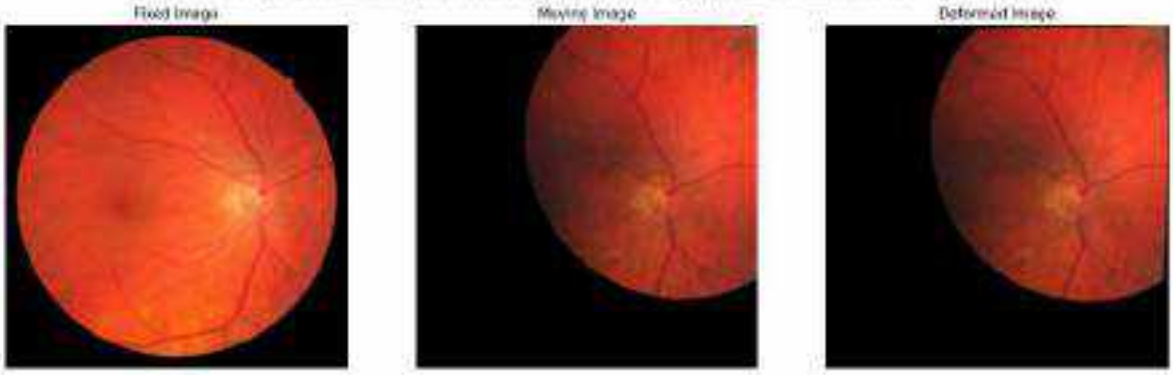

```
Homography Matrix:
[[ 1.04174875e+00 -1.92586062e-02  2.65211448e+02]
 [ 3.55983239e-02  9.75086338e-01 -1.56055107e+02]
 [ 9.57371511e-05 -5.88474611e-05  1.00000000e+00]]
```

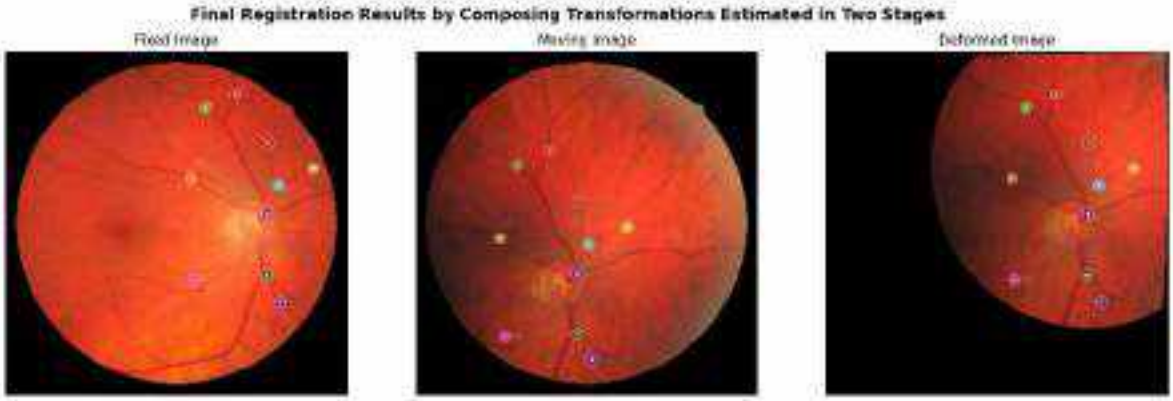



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

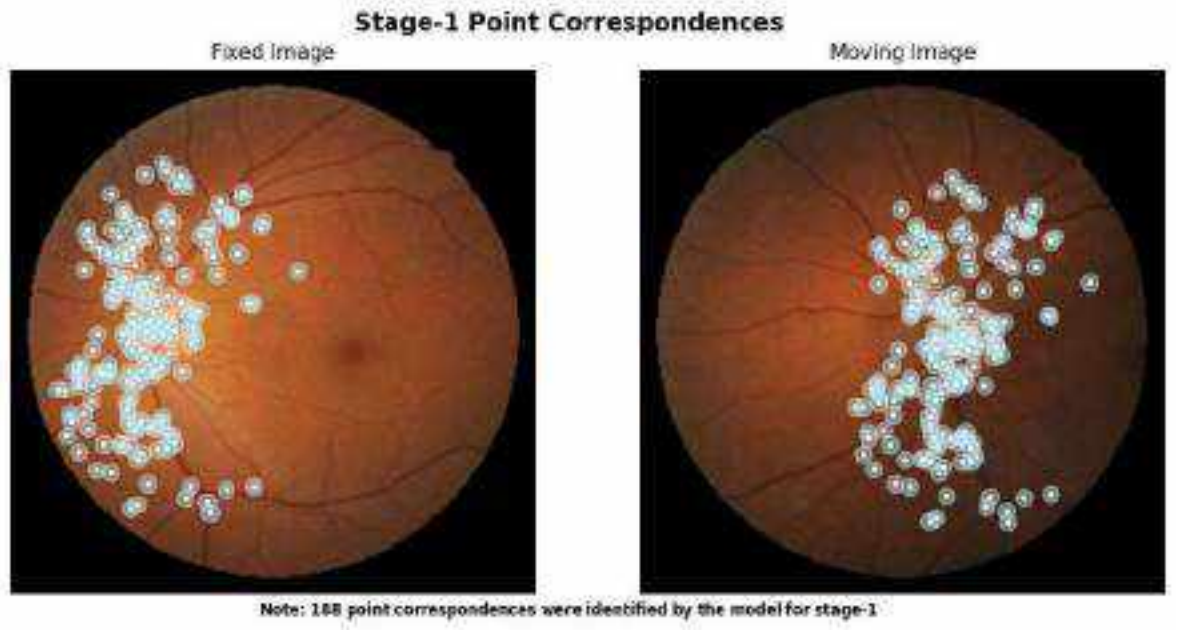


Note: 341 point correspondences were identified by the model for stage-2

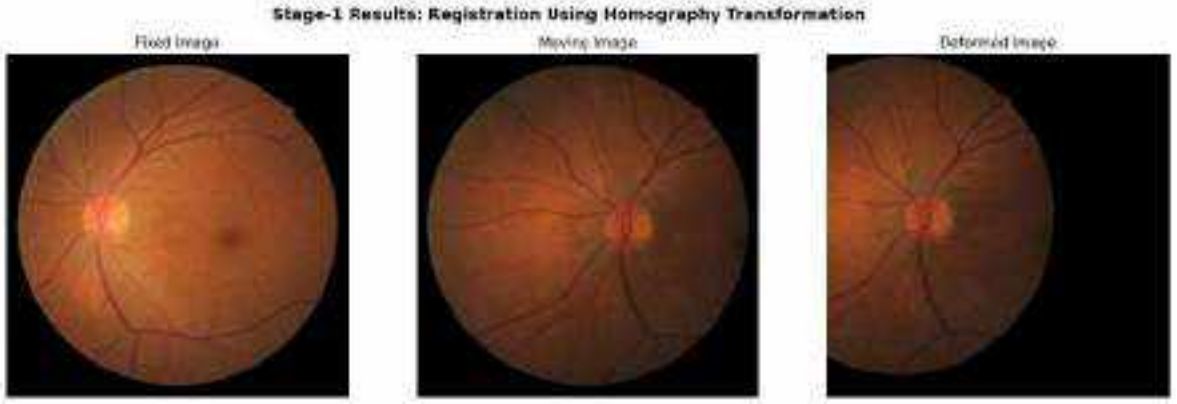




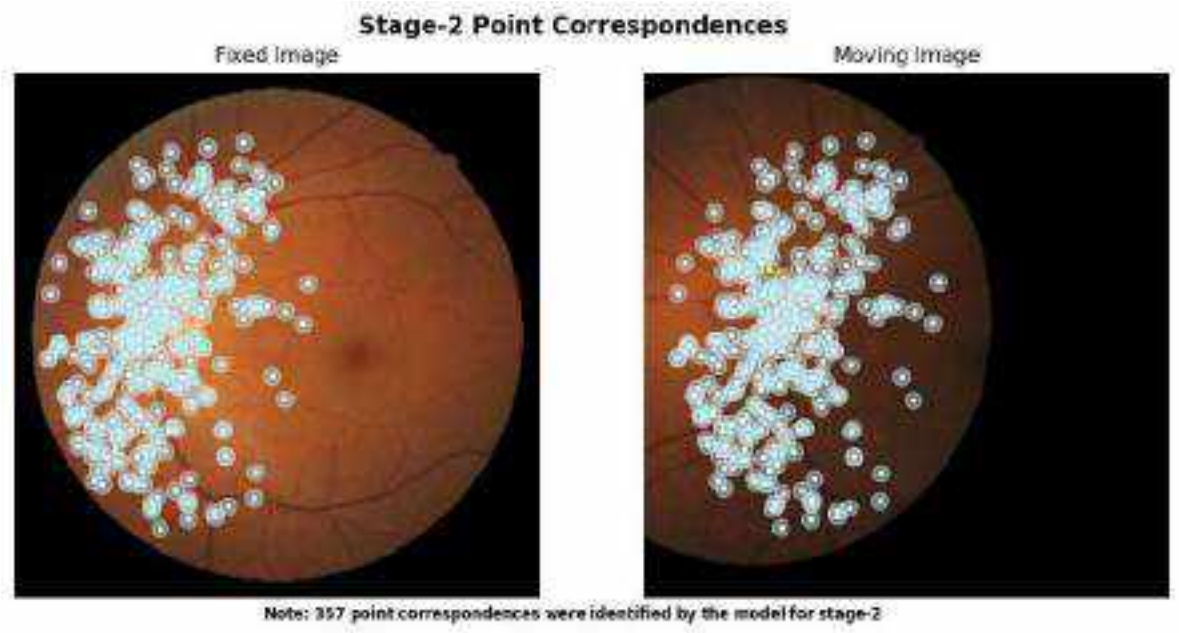
```
Mean Landmark Error for Case 33 Before Registration is 976.0851692197884 pixels
Mean Landmark Error for Case 33 After Registration is 3.4195390849816 53 pixels
Case 34
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P35_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P35_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

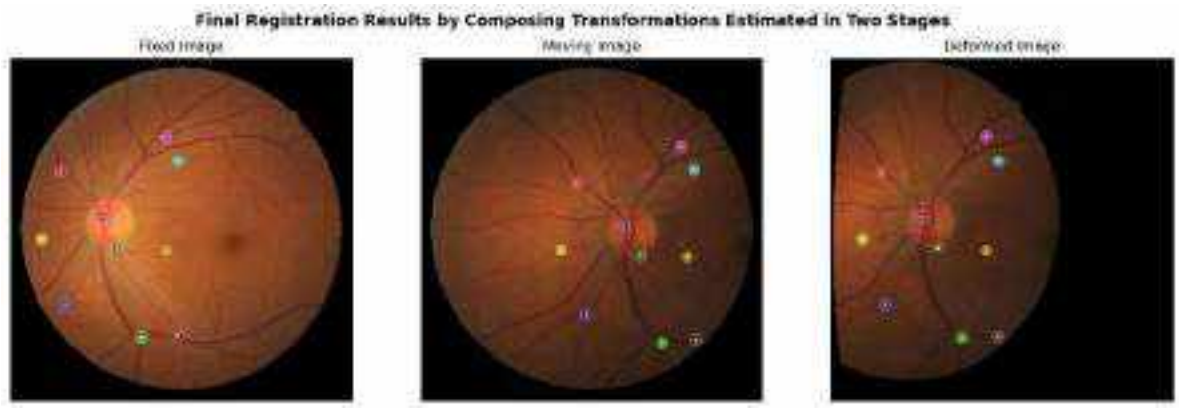



```
Homography Matrix:
[[ 8.58126287e-01 -2.62406577e-02 -2.31360393e+02]
 [-4.73493175e-02  9.15014952e-01  6.27810506e-01]
 [-1.64194523e-04 -2.226881960e-06  1.00000000e+00]]
```

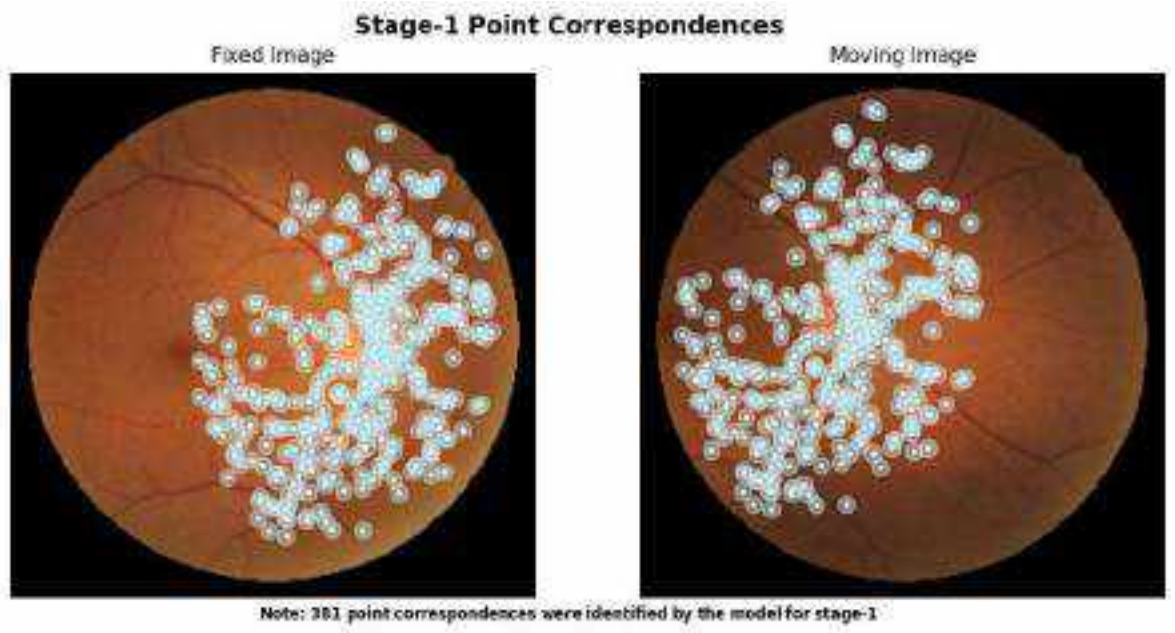



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



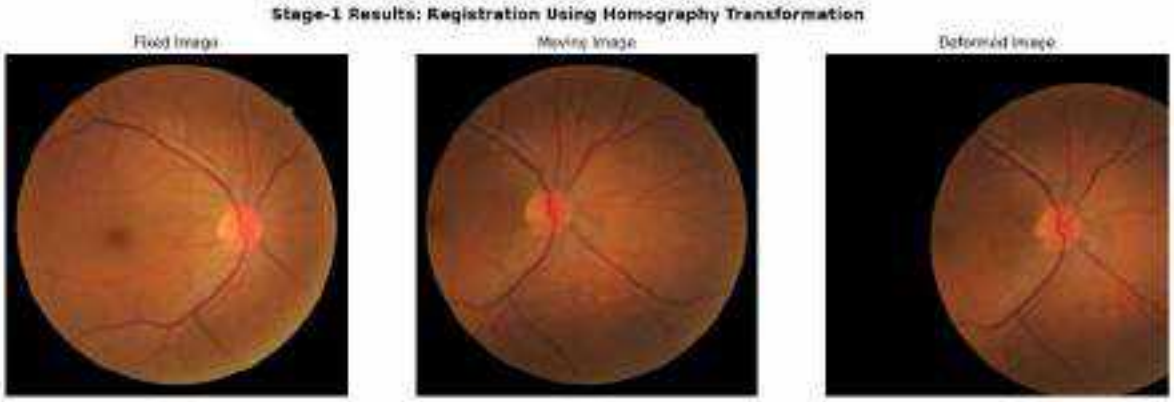



```
Mean Landmark Error for Case 34 Before Registration is 932.9108132213235 pixels
Mean Landmark Error for Case 34 After Registration is 3.1296377783829747 pixels
Case 35
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P36_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P36_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

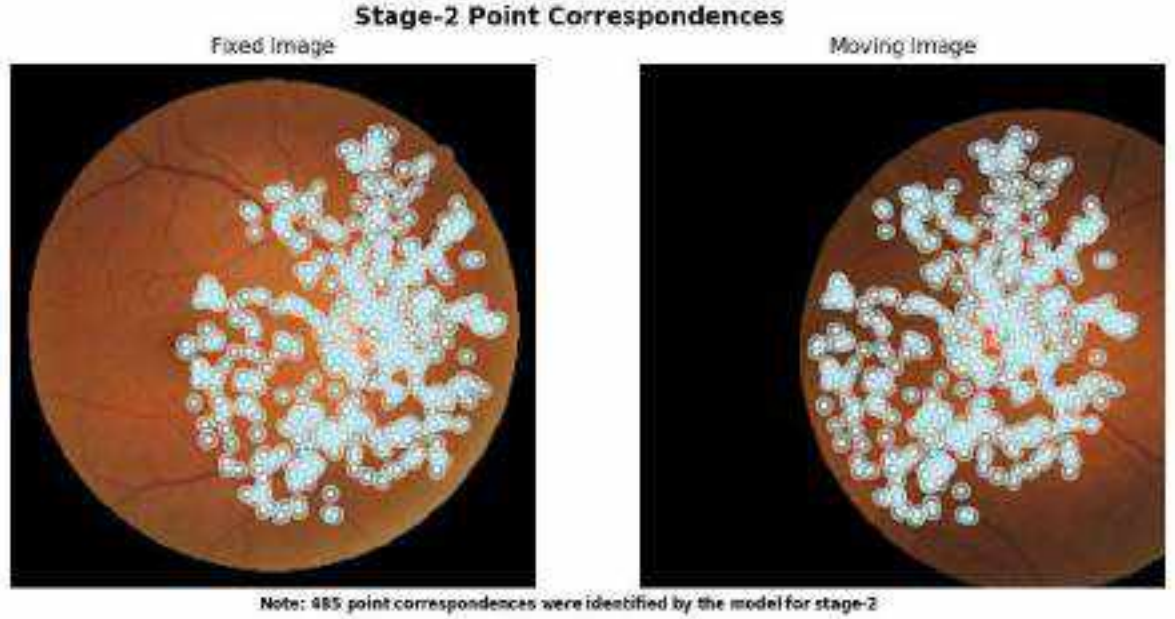


Note: 381 point correspondences were identified by the model for stage-1

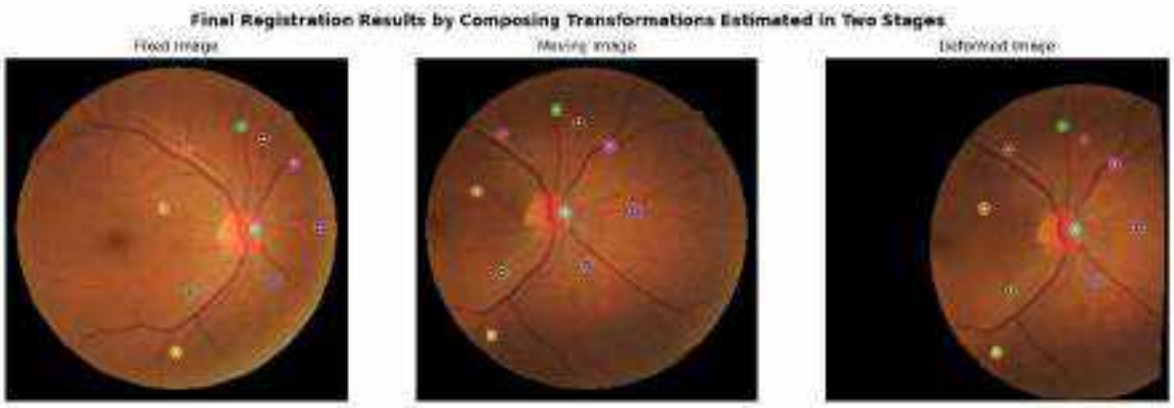
```
Homography Matrix:
[[1.15611445e+00 7.80386636e-03 2.44819324e+02]
 [8.12502735e-02 1.06495101e+00 1.82068738e+01]
 [1.61467350e-04 8.40085693e-06 1.00000000e+00]]
```



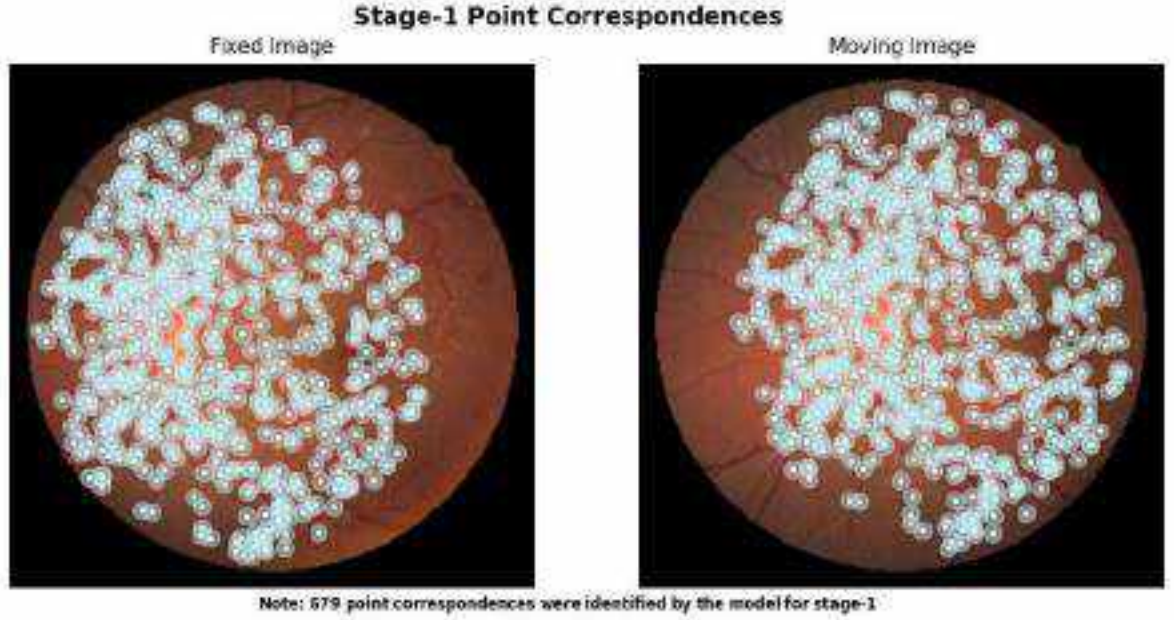

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



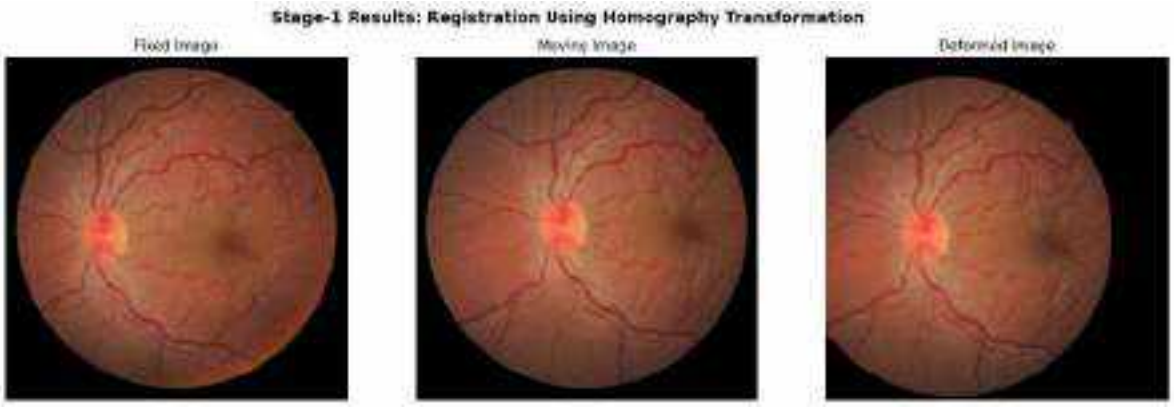
Note: 485 point correspondences were identified by the model for stage-2



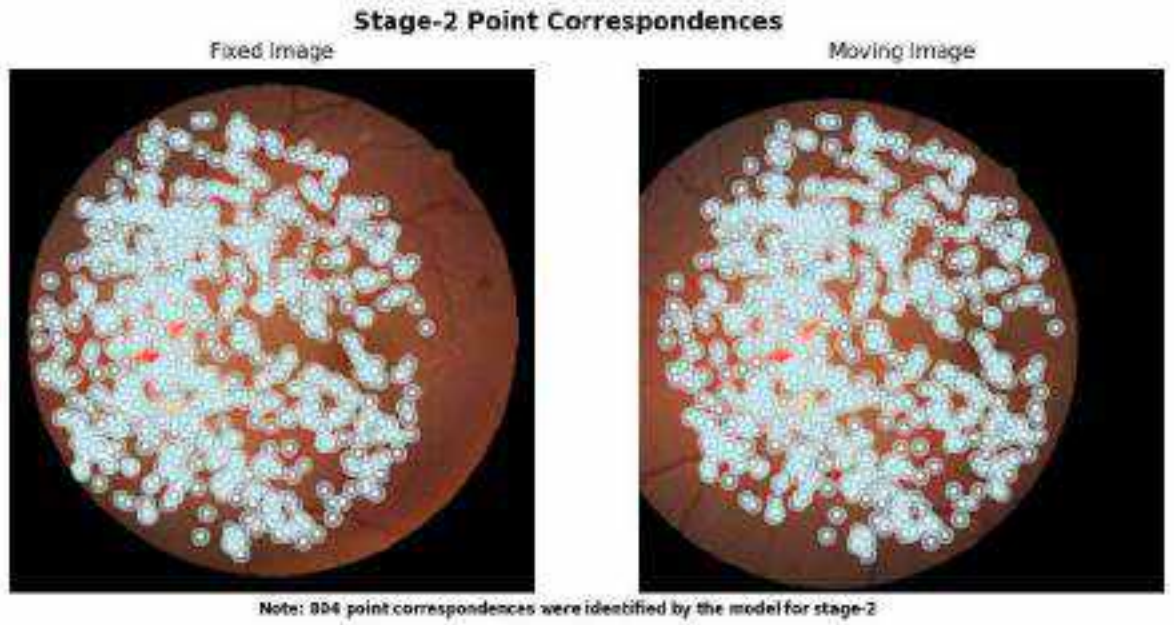

```
Mean Landmark Error for Case 35 Before Registration is 835.4558619190868 pixels
Mean Landmark Error for Case 35 After Registration is 3.0658040417685704 pixels
Case 36
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P37_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P37_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



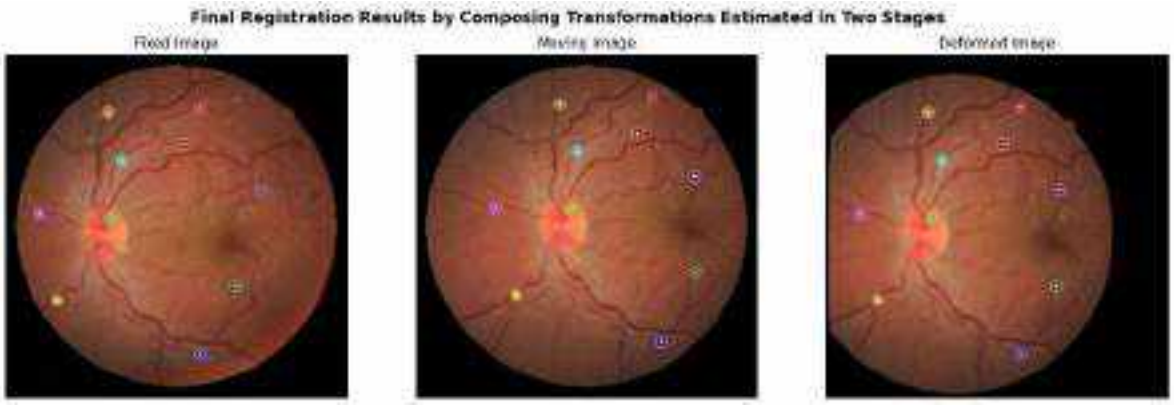

```
Homography Matrix:
[[ 9.48623755e-01 -3.21739441e-02 -9.91396910e+01]
 [ 8.26998857e-03  9.76798661e-01  2.85435745e+01]
 [-6.17947963e-05  1.17714792e-05  1.00000000e+00]]
```



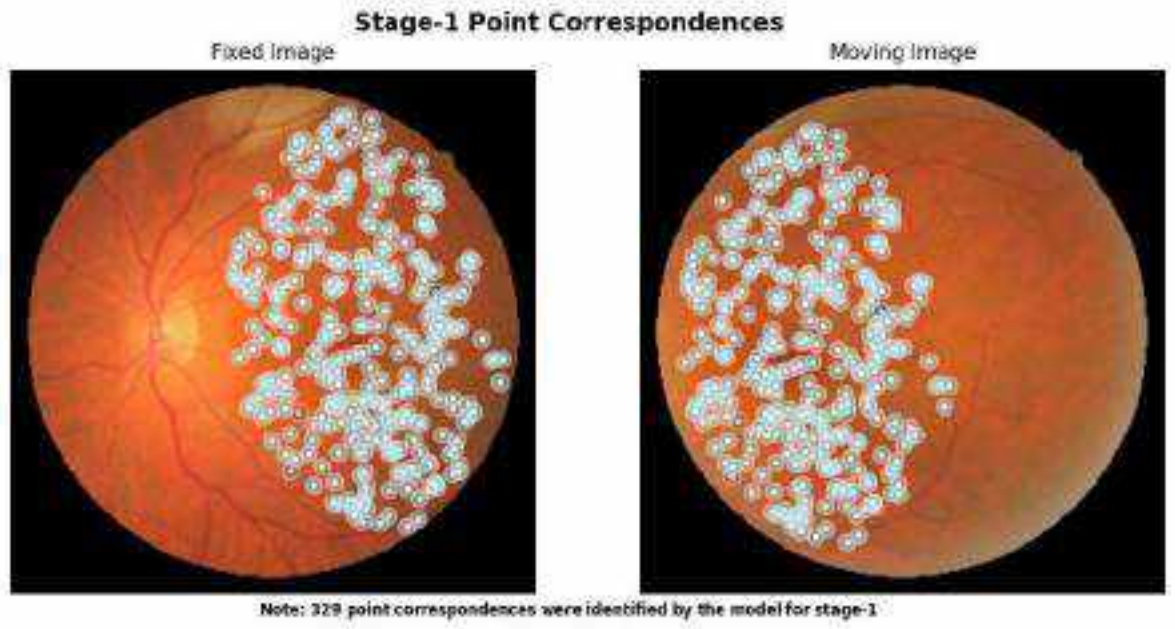

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 804 point correspondences were identified by the model for stage-2

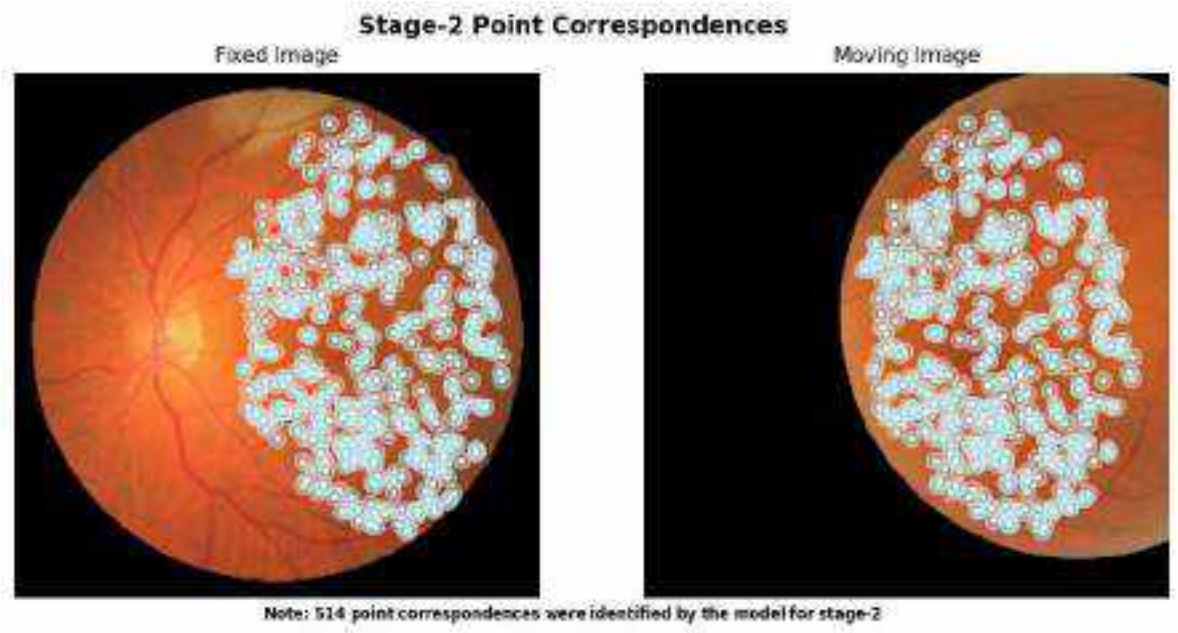


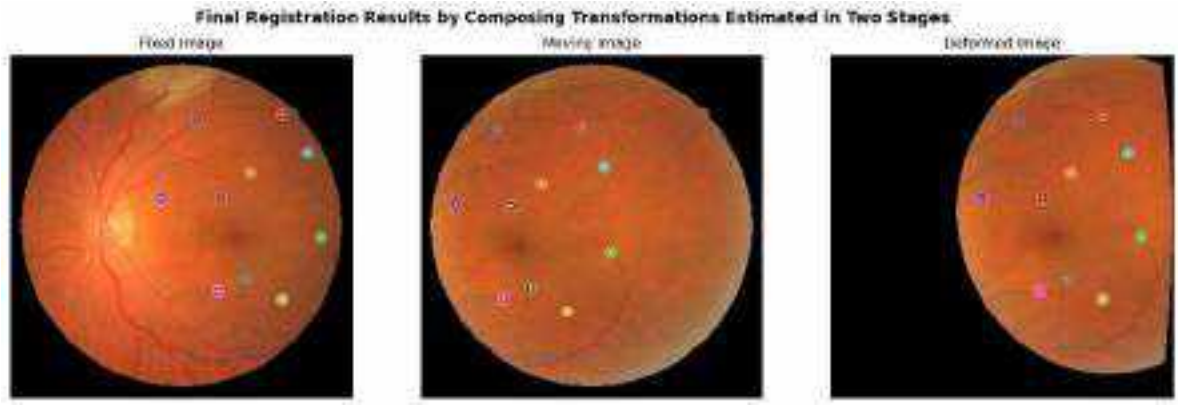
```
Mean Landmark Error for Case 36 Before Registration is 385.1129366309333 pixels
Mean Landmark Error for Case 36 After Registration is 20.0043989316196 65 pixels
Case 37
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P38_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P38_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



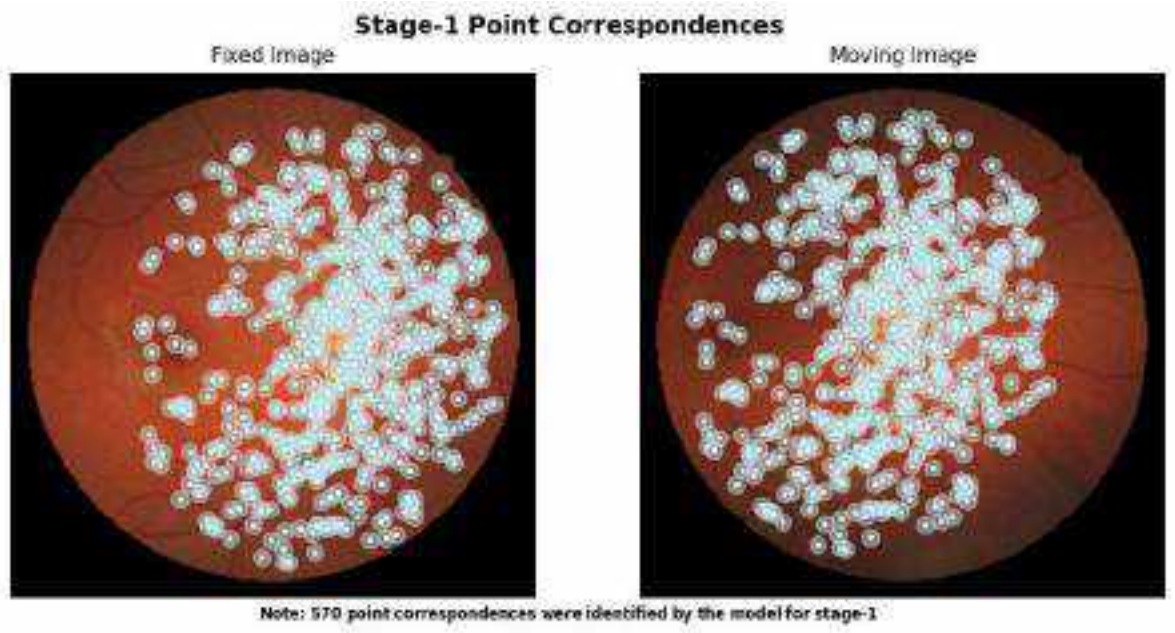

```
Homography Matrix:
[[ 1.11101964e+00  7.66664810e-02  2.78781767e+02]
 [-1.96824207e-02  1.05210353e+00 -2.79036354e+01]
 [ 1.15651527e-04  8.34561759e-06  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



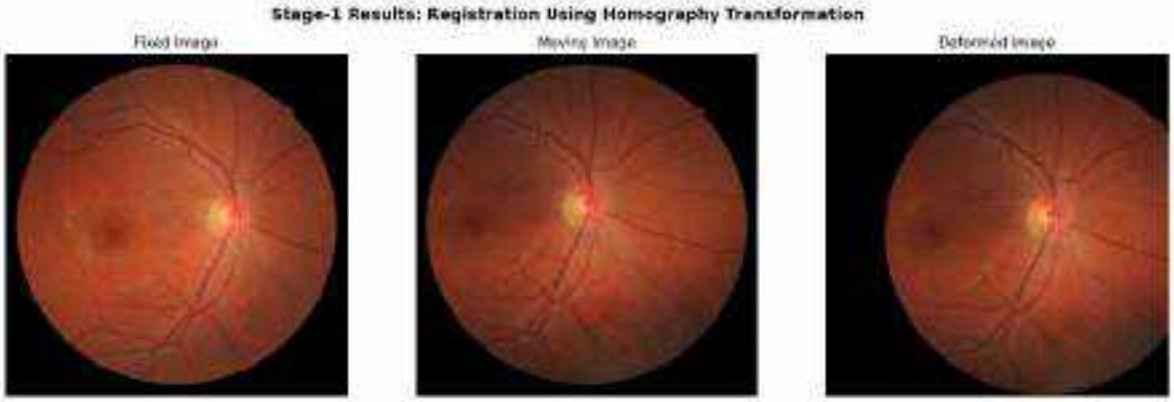



```
Mean Landmark Error for Case 37 Before Registration is 1014.8722169521454 pixels
Mean Landmark Error for Case 37 After Registration is 2.562126395582213 pixels
Case 38
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P39_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P39_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

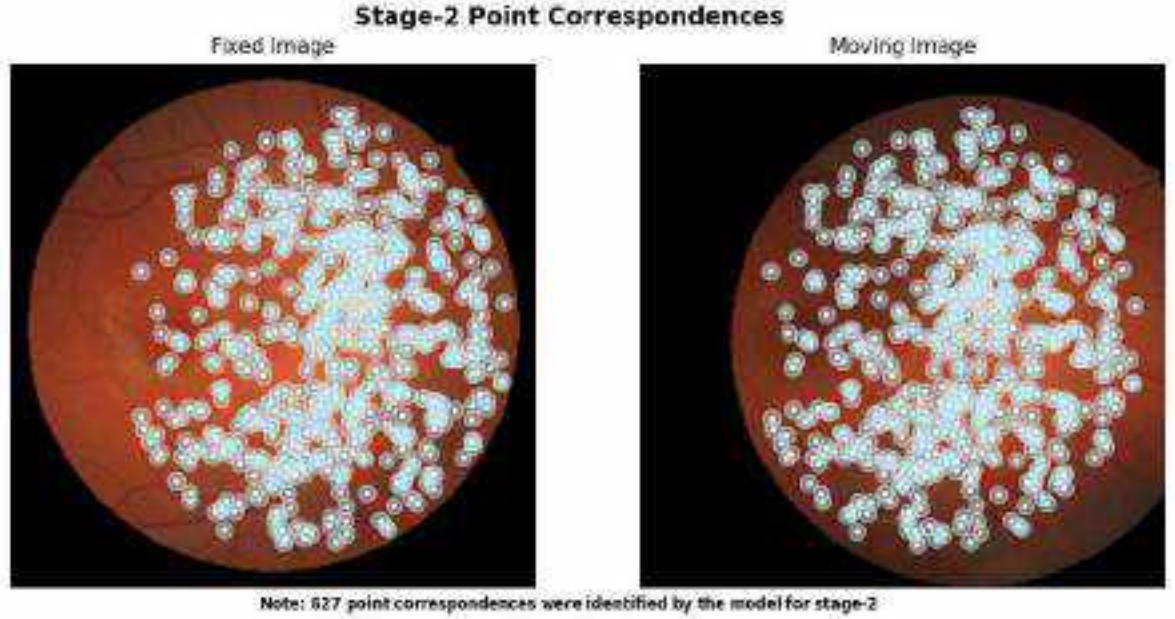


Note: 570 point correspondences were identified by the model for stage-1

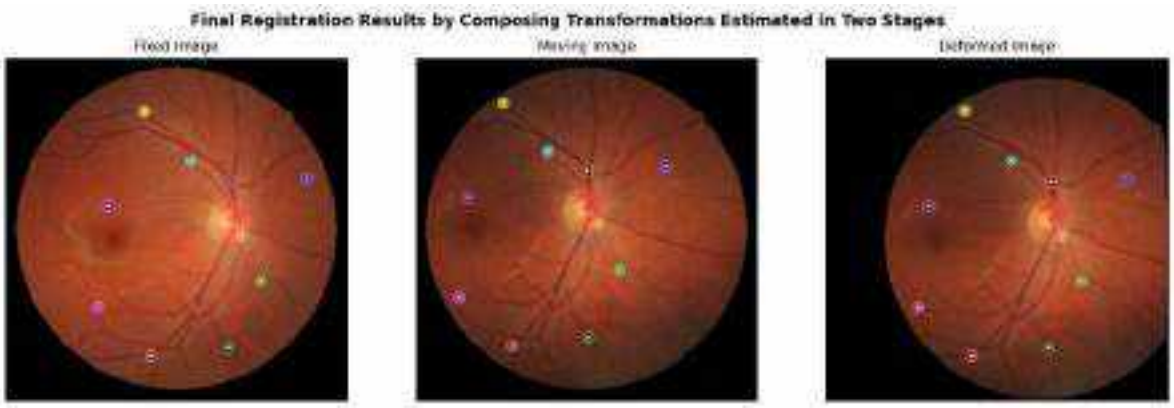
```
Homography Matrix:
[[ 1.08483265e+00 -2.24189873e-03  1.29893546e+02]
 [ 5.32200890e-02  1.05150638e+00  3.01862370e+00]
 [ 8.02594458e-05  1.65888680e-05  1.00000000e+00]]
```



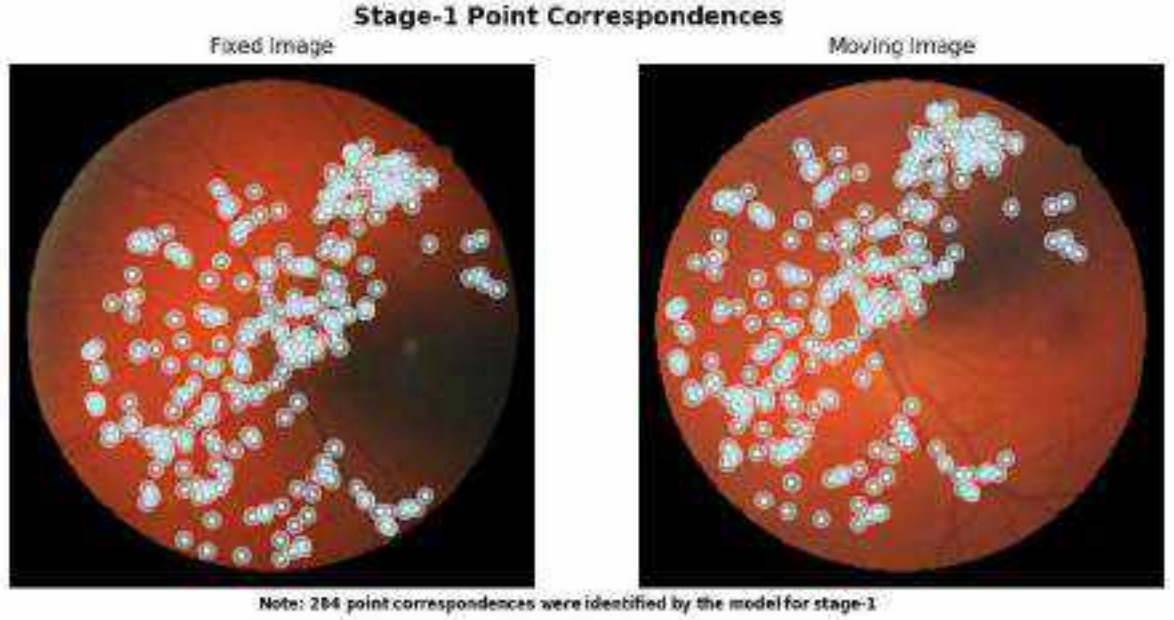

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



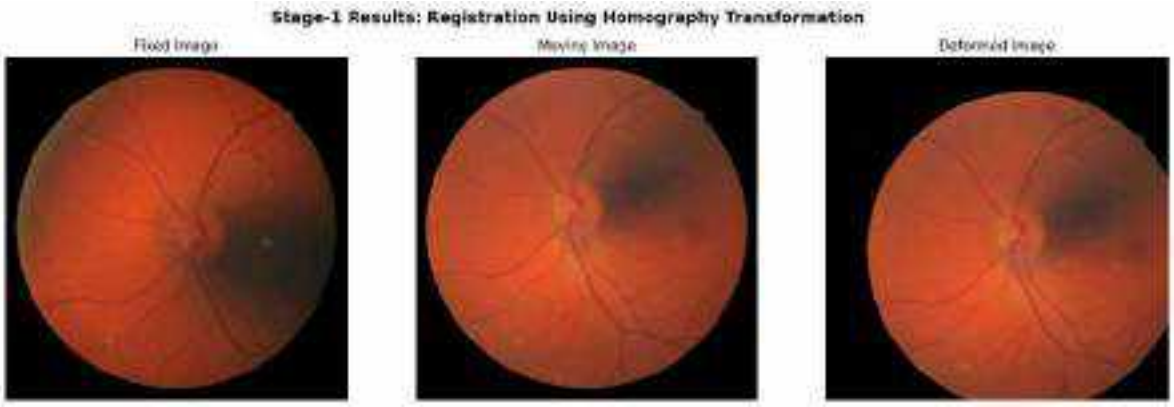
Note: 627 point correspondences were identified by the model for stage-2



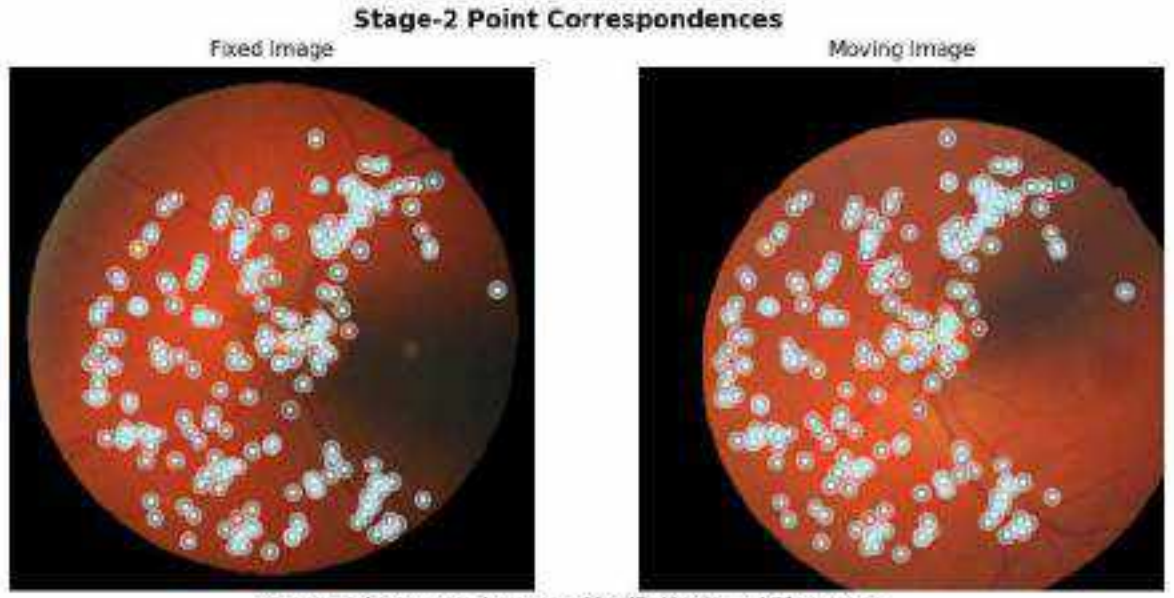

```
Mean Landmark Error for Case 38 Before Registration is 445.32183299843 67 pixels
Mean Landmark Error for Case 38 After Registration is 2.3916435597177887 pixels
Case 39
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P40_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P40_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



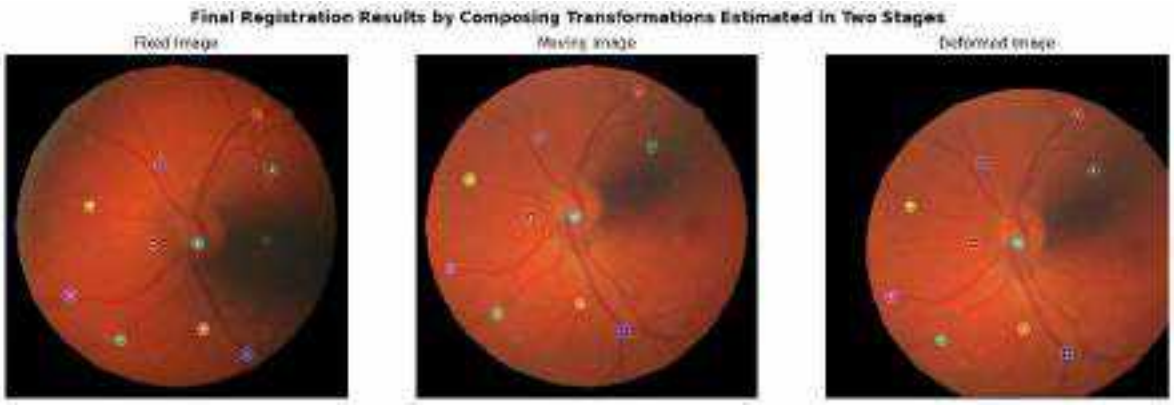

```
Homography Matrix:
[[1.86127138e+00 3.55595183e-02 6.45393578e+01]
 [1.19234733e-02 1.05592620e+00 5.86132497e+01]
 [4.95522587e-05 3.82820010e-05 1.00000000e+00]]
```



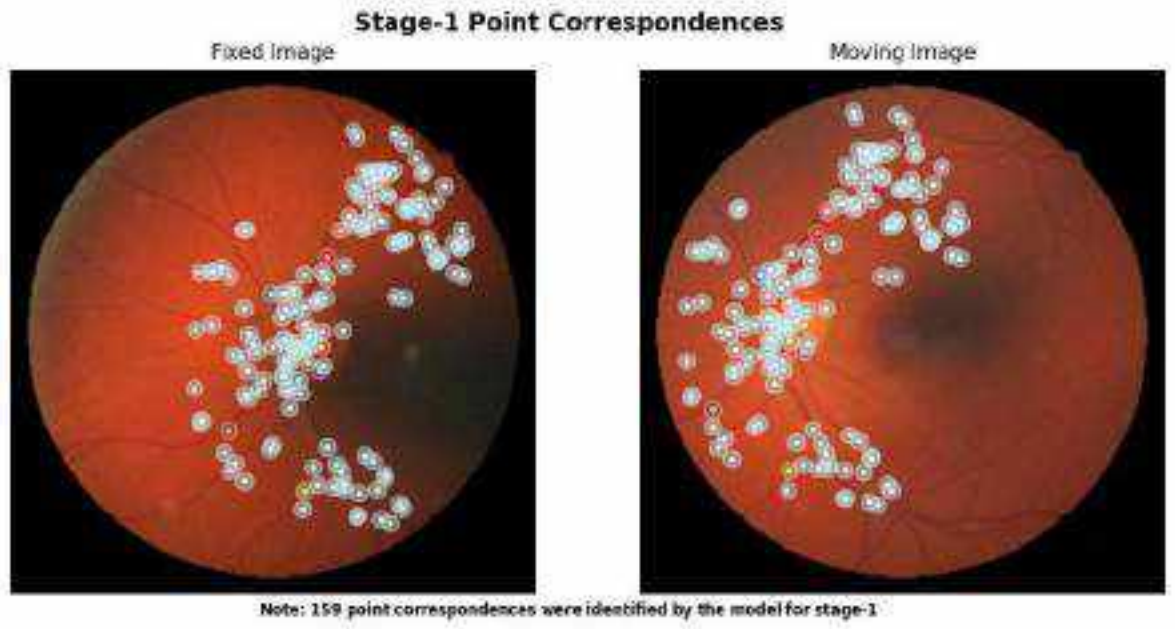

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



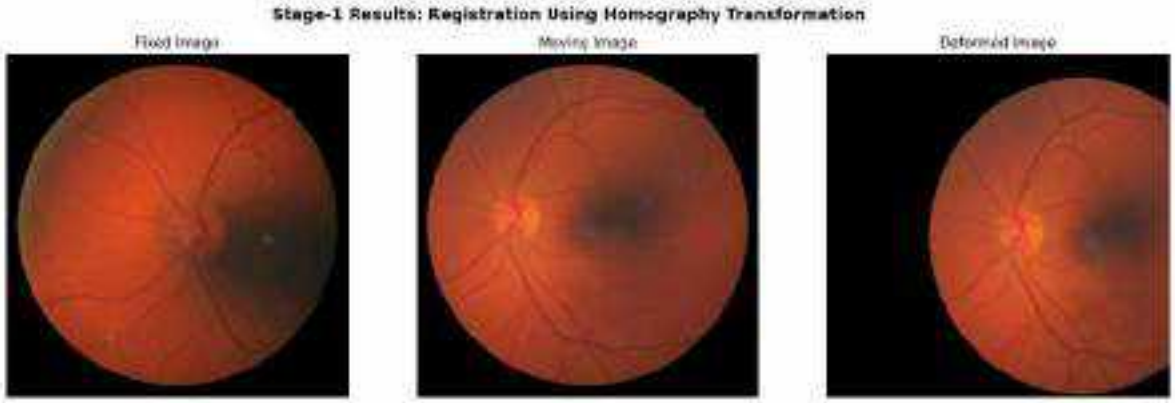
Note: 231 point correspondences were identified by the model for stage-2



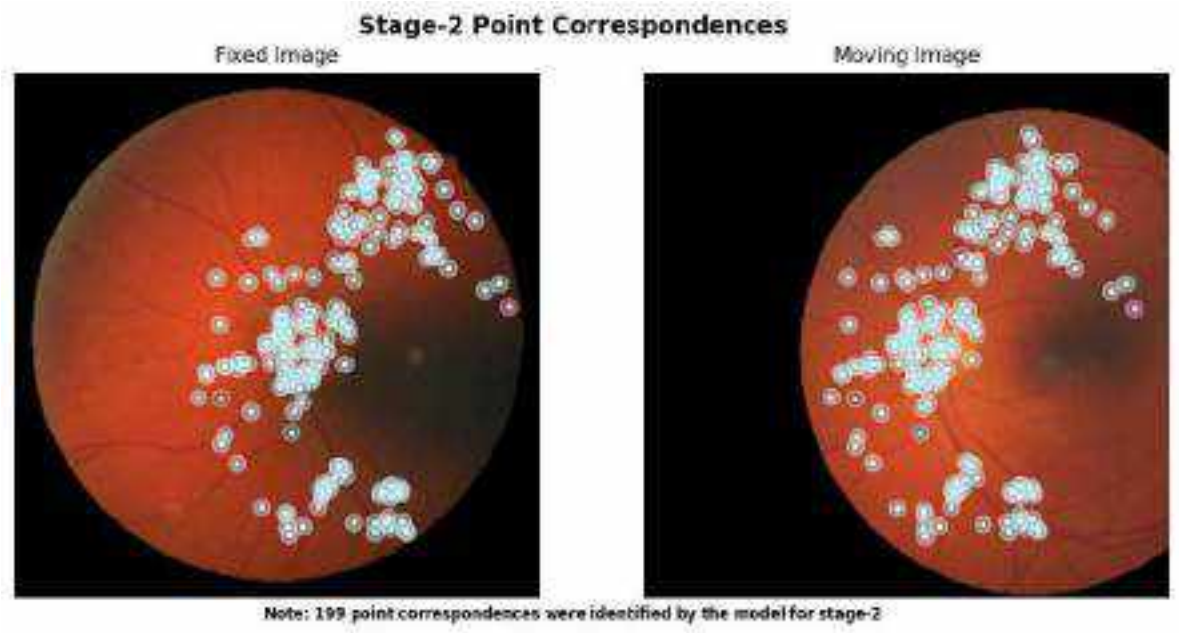
```
Mean Landmark Error for Case 39 Before Registration is 346.32638796321316 pixels
Mean Landmark Error for Case 39 After Registration is 3.5503758937367174 pixels
Case 40
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P41_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P41_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

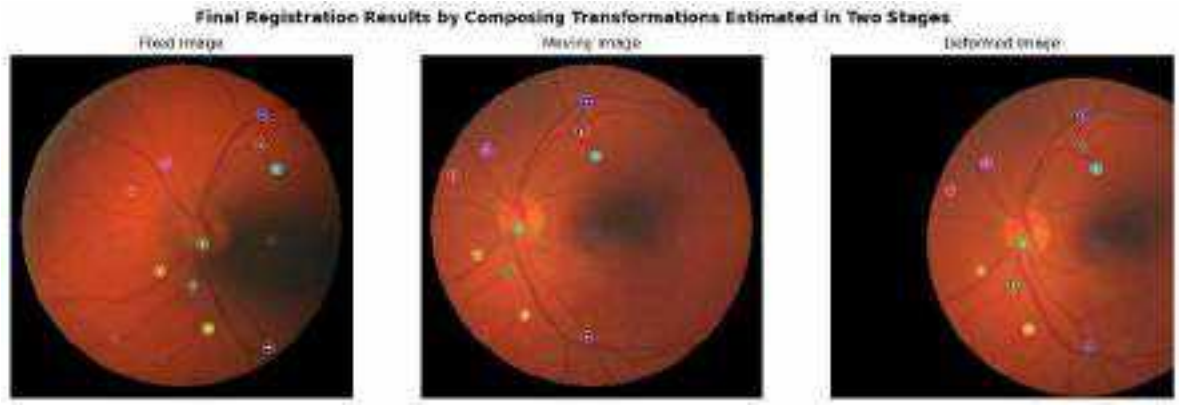



```
Homography Matrix:
[[1.11011572e+00 3.97464201e-02 2.27464248e+02]
 [3.86215193e-02 1.00762494e+00 1.99263497e+01]
 [1.37691271e-04 2.31283161e-05 1.00000000e+00]]
```

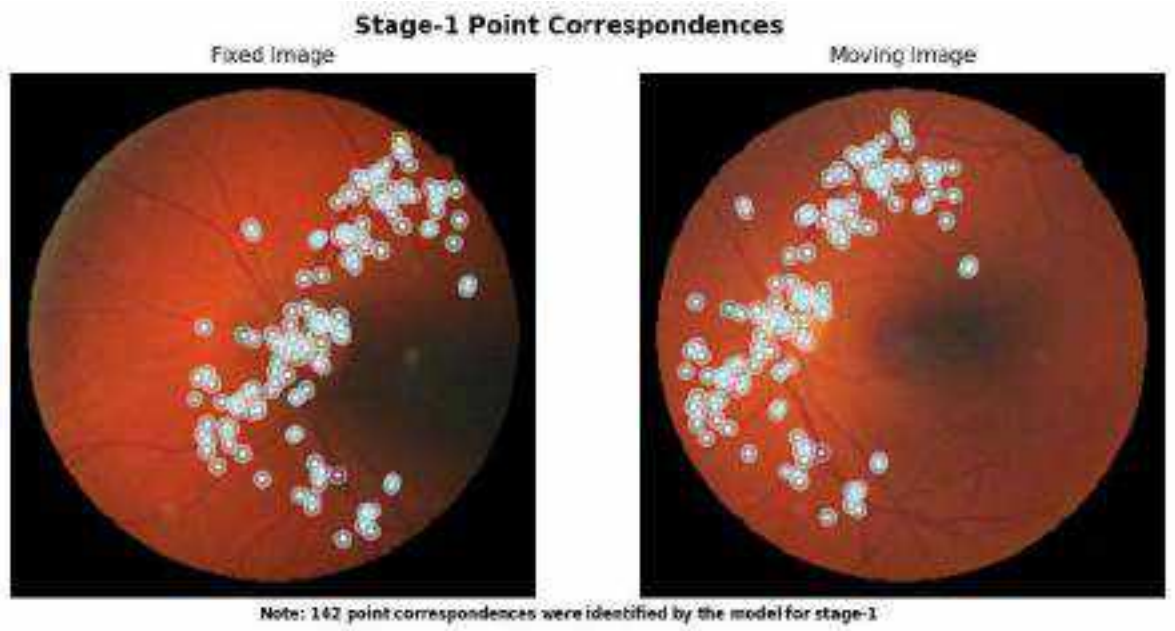



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





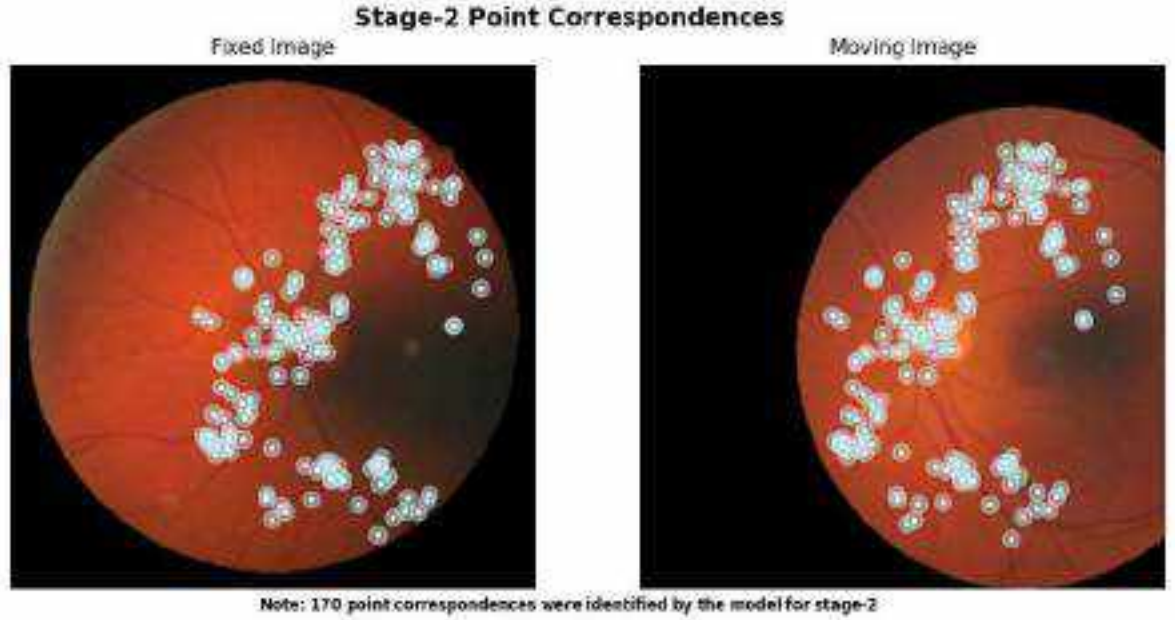

```
Mean Landmark Error for Case 40 Before Registration is 785.4529070187097 pixels
Mean Landmark Error for Case 40 After Registration is 3.80039609731001 pixels
Case 41
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P42_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P42_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



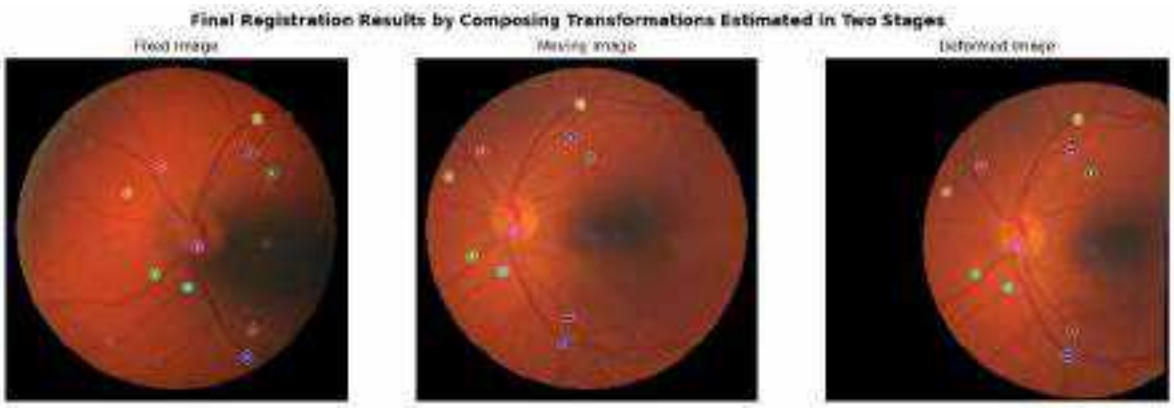
Note: 142 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.12010851e+00  1.82029866e-02  2.34340965e+02]
 [ 4.88117034e-02  1.05411917e+00  2.55913751e+01]
 [ 1.67867071e-04 -1.13681024e-06  1.00000000e+00]]
```

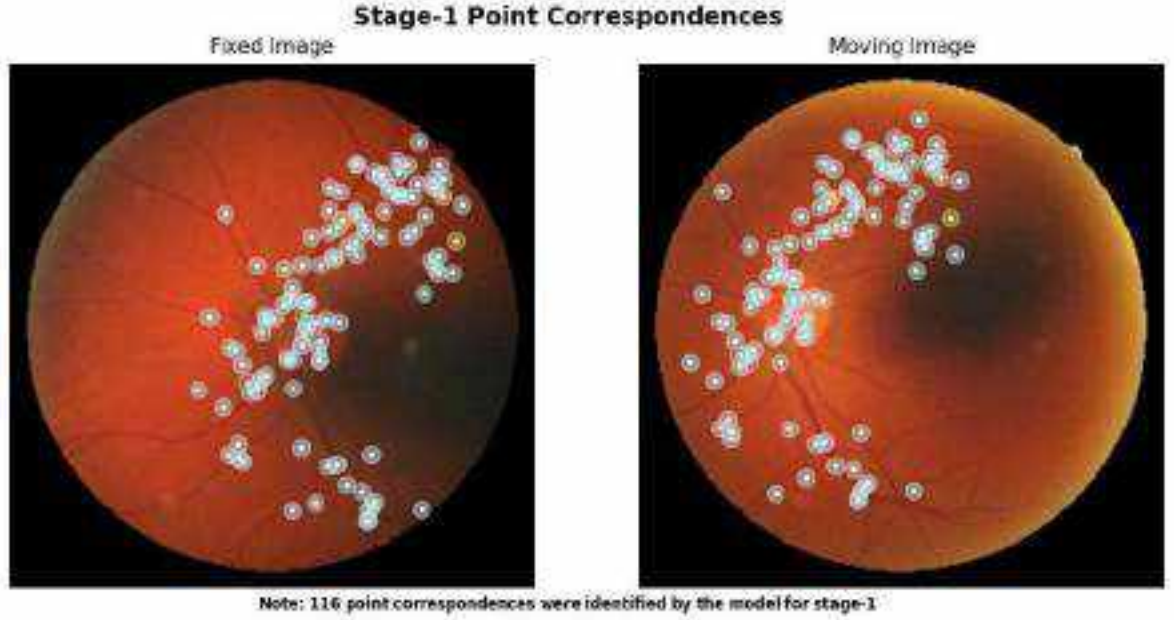

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



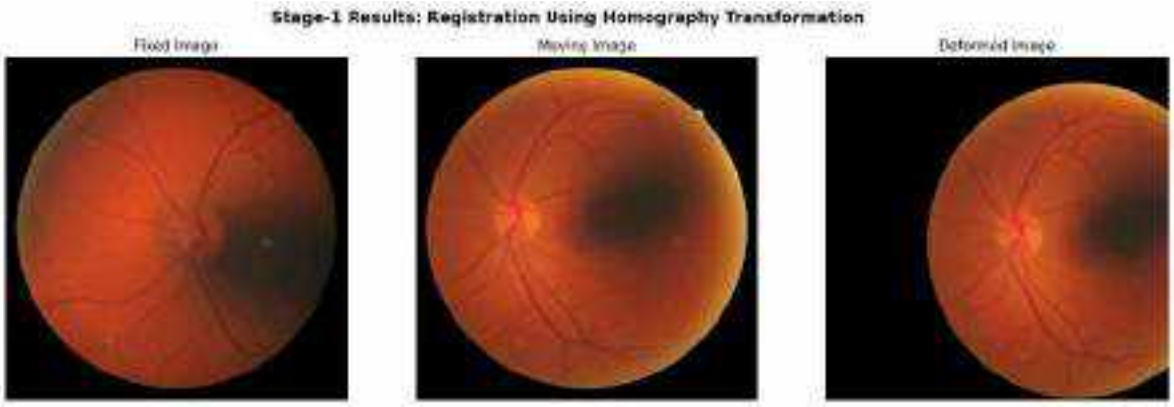
Note: 170 point correspondences were identified by the model for stage-2



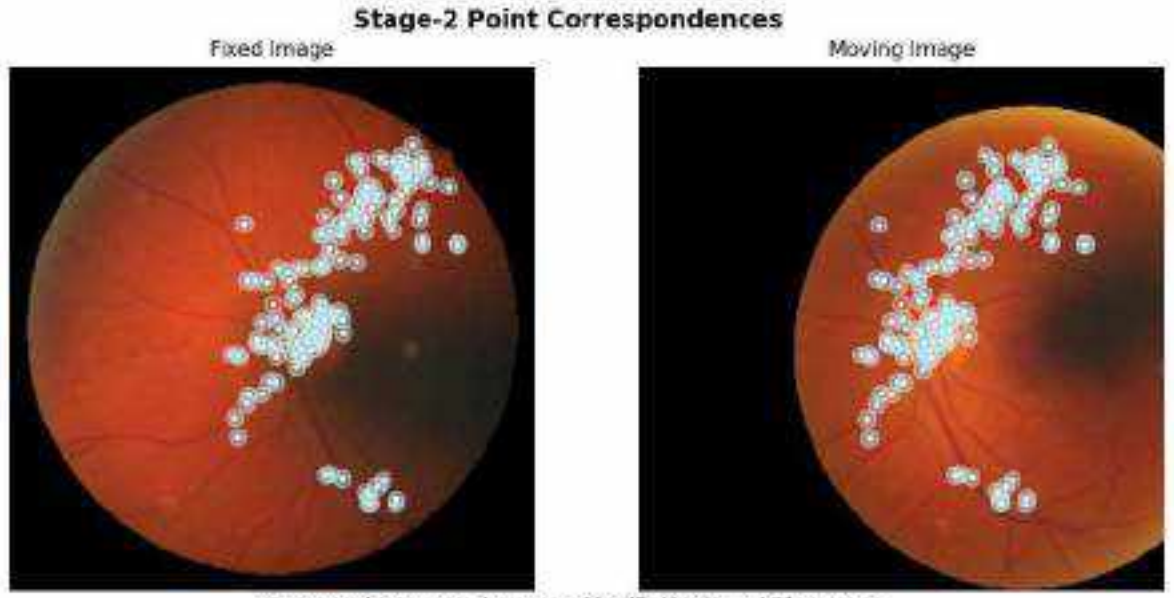

```
Mean Landmark Error for Case 41 Before Registration is 791.5831876447837 pixels
Mean Landmark Error for Case 41 After Registration is 5.984962803063208 pixels
Case 42
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P43_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P43_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



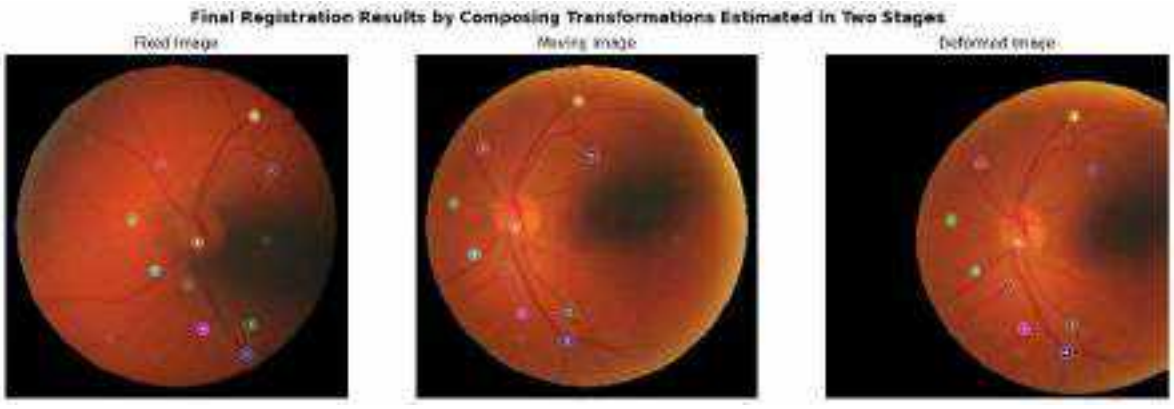

```
Homography Matrix:
[[1.10132904e+00 3.13471213e-02 2.30091856e+02]
 [3.80341166e-02 1.06899902e+00 2.47511325e+01]
 [1.26543823e-04 3.19302488e-05 1.00000000e+00]]
```



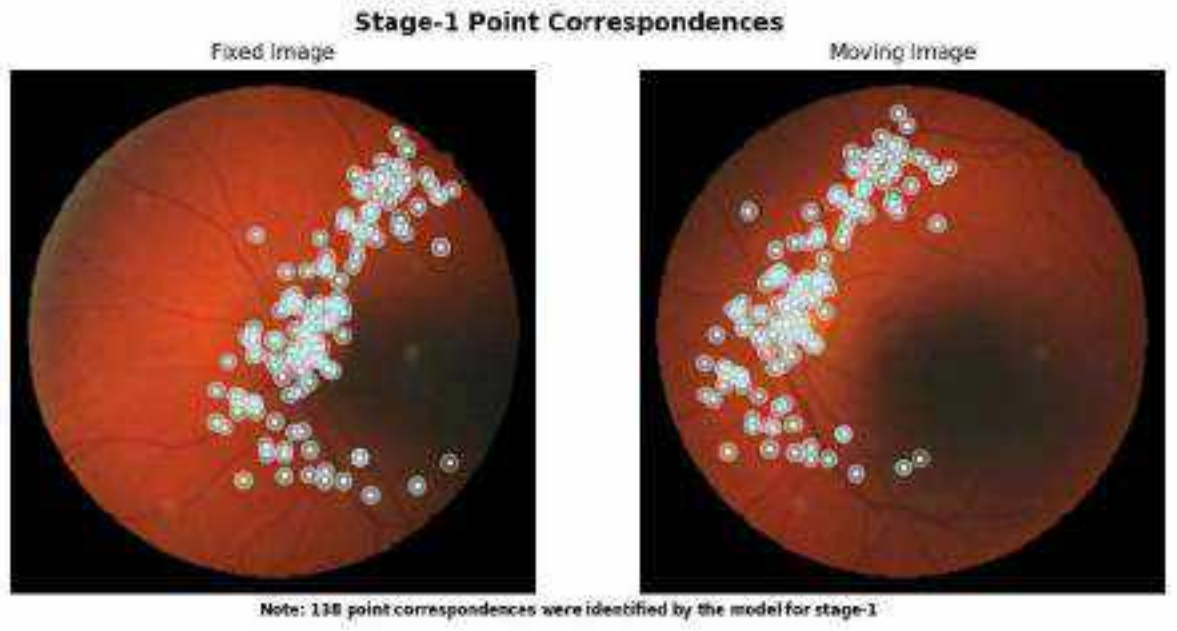

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 143 point correspondences were identified by the model for stage-2

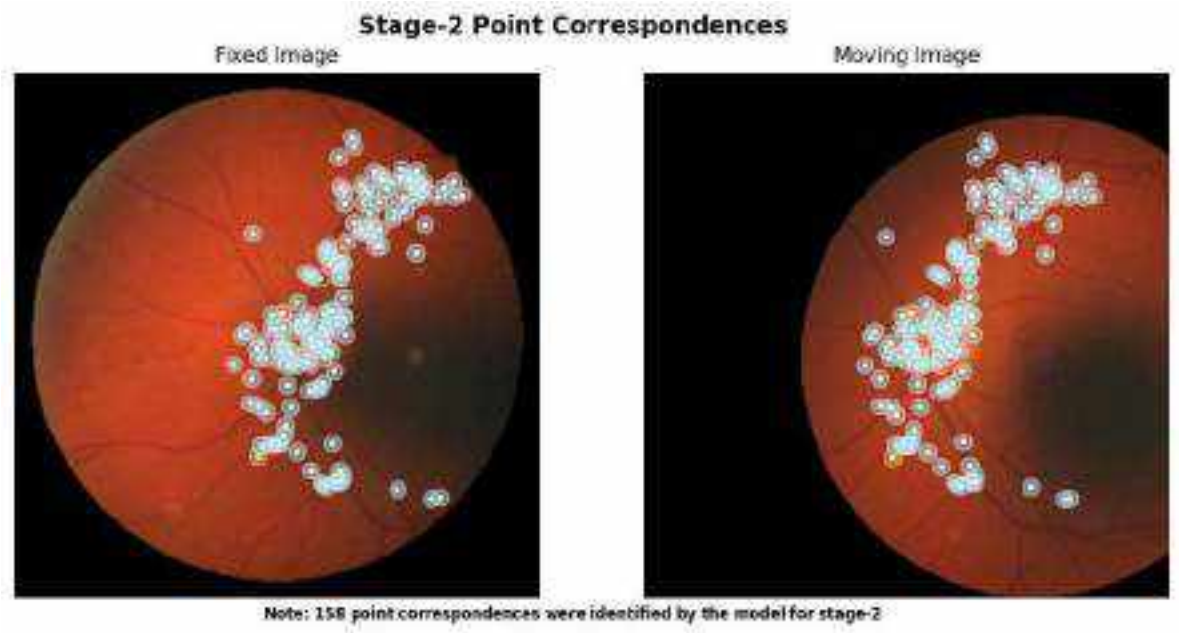


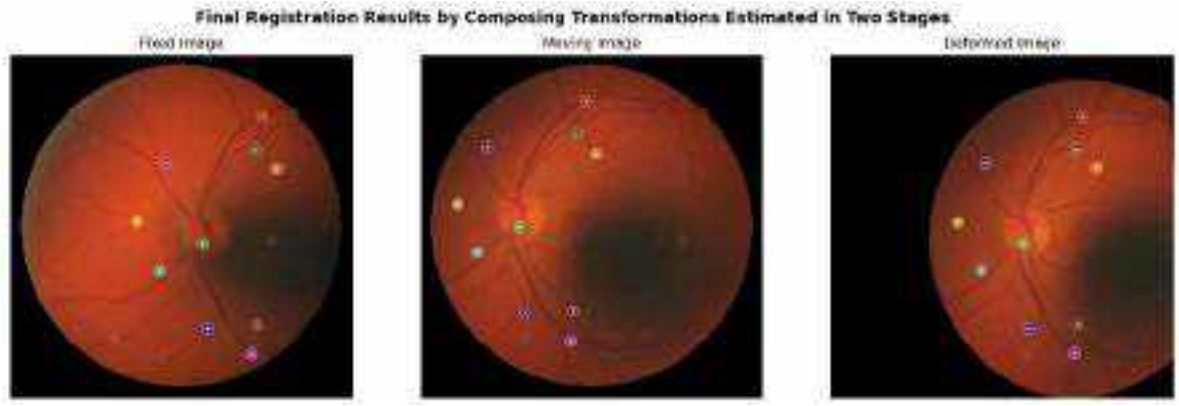
```
Mean Landmark Error for Case 42 Before Registration is 779.7105743640745 pixels
Mean Landmark Error for Case 42 After Registration is 5.5865421658841032 pixels
Case 43
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P44_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P44_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



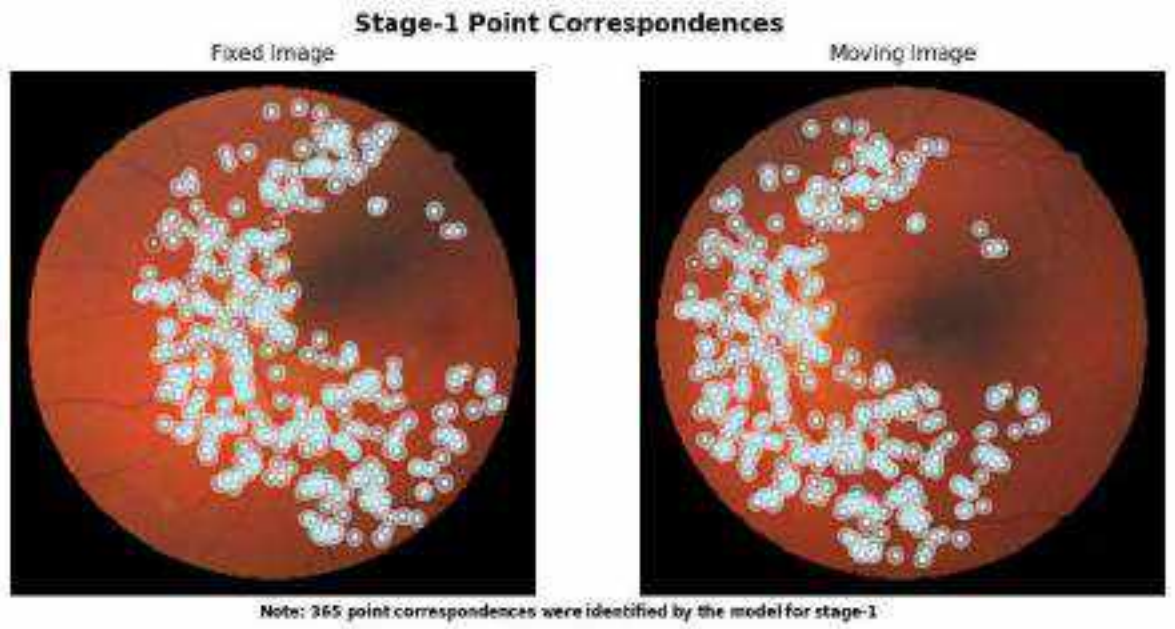

```
Homography Matrix:
[[ 1.08139971e+00  2.91249914e-02  2.29993219e+02]
 [ 2.31419011e-03  1.01762909e+00  4.65871868e+01]
 [ 1.14879810e-04 -1.31898176e-05  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



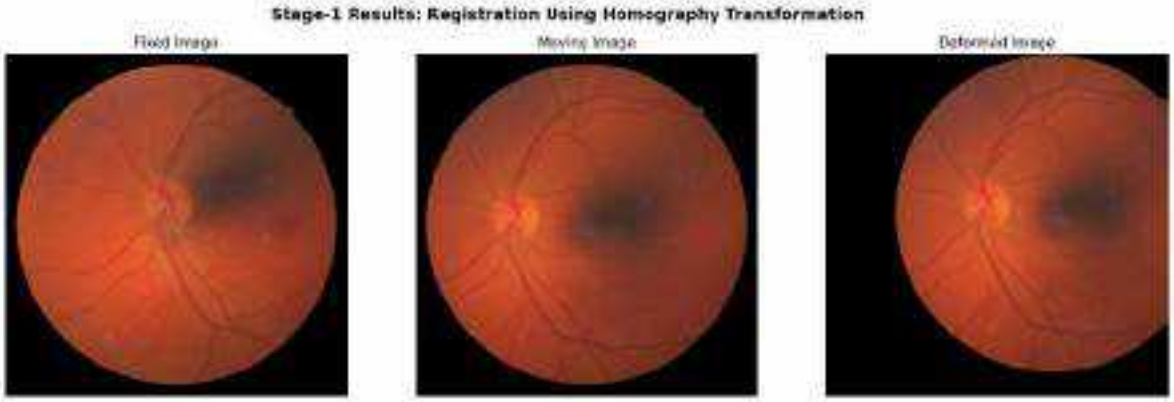



```
Mean Landmark Error for Case 43 Before Registration is 785.1280295134127 pixels
Mean Landmark Error for Case 43 After Registration is 8.2144081896115222 pixels
Case 44
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P45_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P45_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

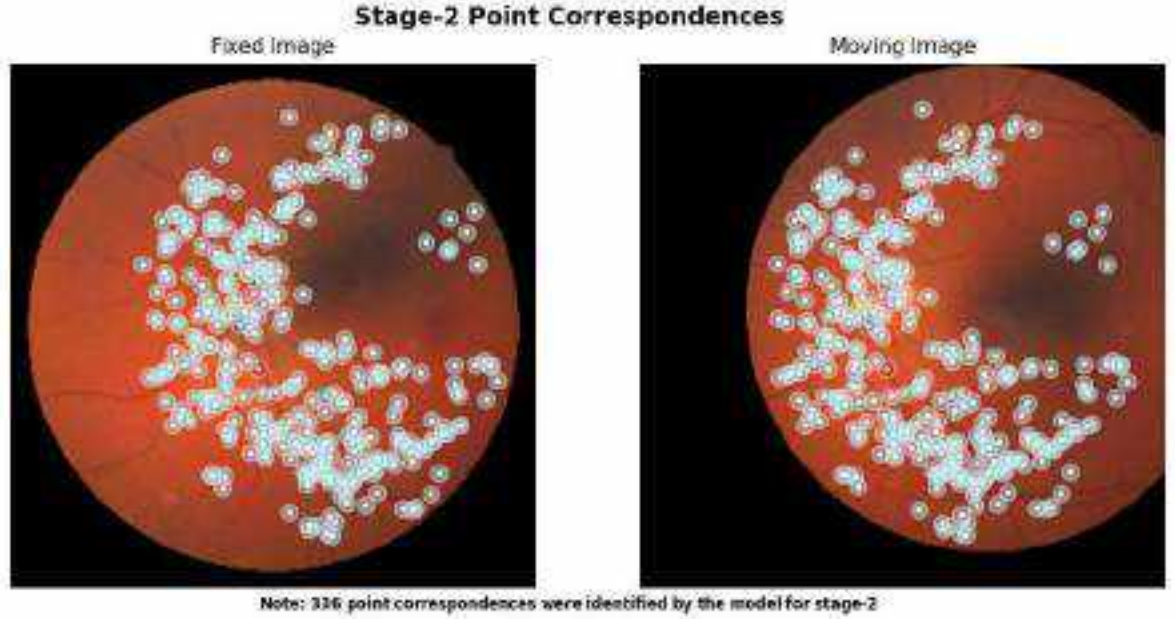


Note: 365 point correspondences were identified by the model for stage-1

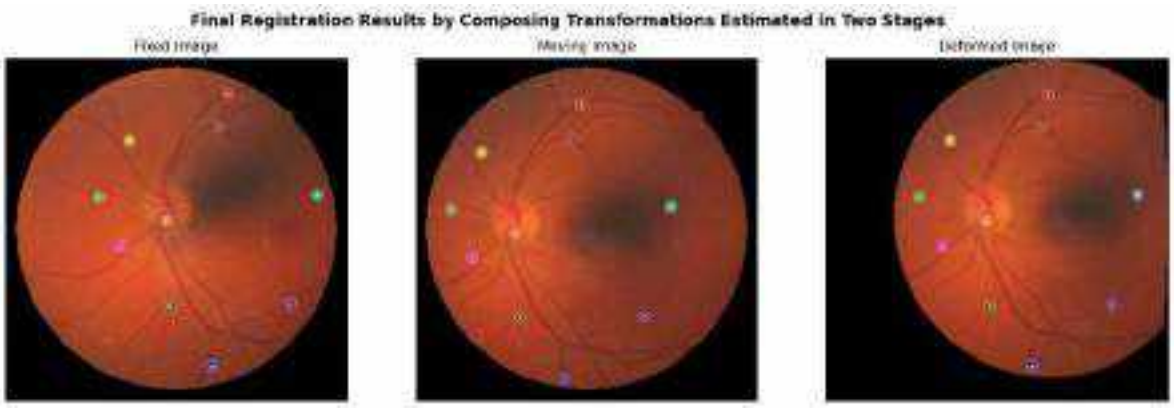
```
Homography Matrix:
[[ 1.04660853e+00 -7.86313003e-04  1.55688265e+02]
 [ 2.90831893e-02  1.01393762e+00 -4.00327043e+01]
 [ 6.78498428e-05 -9.08579656e-06  1.00000000e+00]]
```



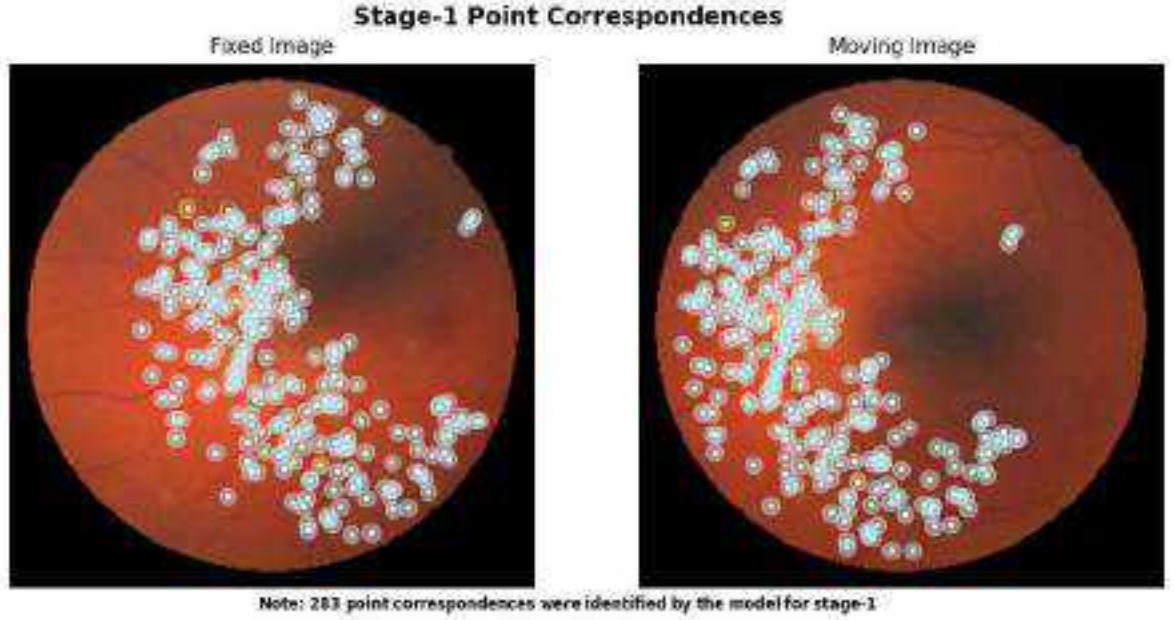

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 336 point correspondences were identified by the model for stage-2

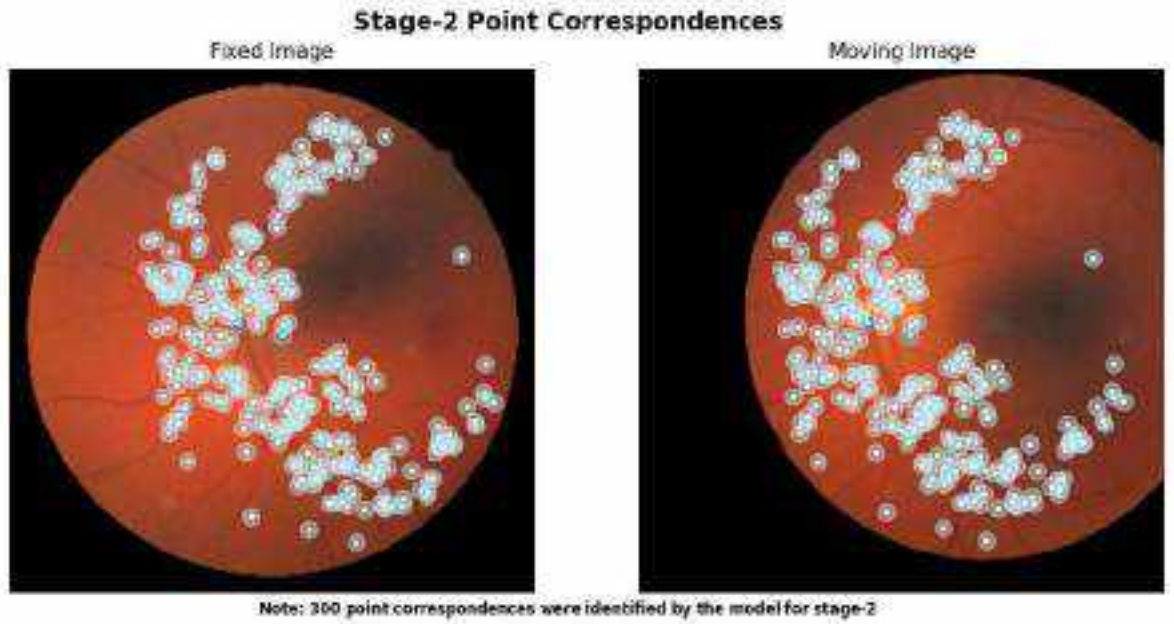



```
Mean Landmark Error for Case 44 Before Registration is 509.92136010471734 pixels
Mean Landmark Error for Case 44 After Registration is 2.70583735275279 pixels
Case 45
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P46_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P46_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

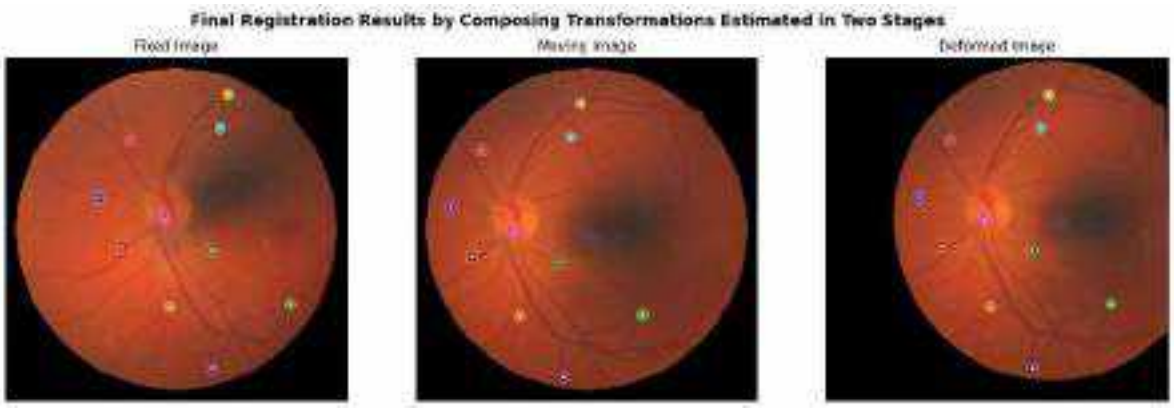



```
Homography Matrix:
[[ 1.06585539e+00 -6.59414927e-03  1.55759582e+02]
 [ 4.09010461e-02  1.01394156e+00 -3.78361764e+01]
 [ 9.21754092e-05 -1.78957771e-05  1.00000000e+00]]
```

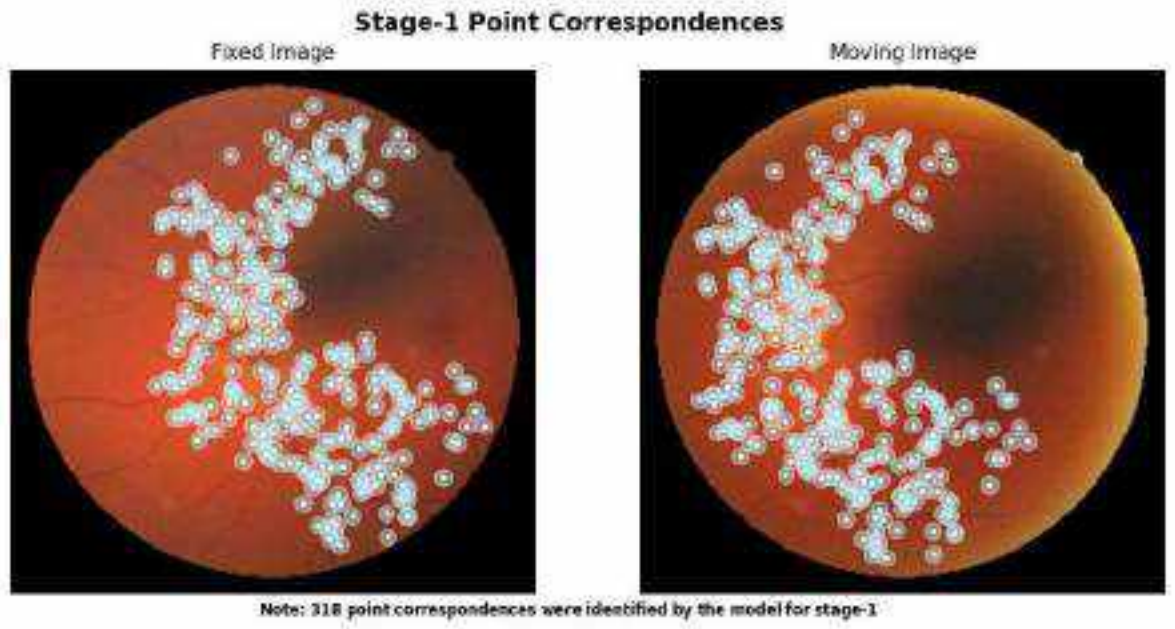

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



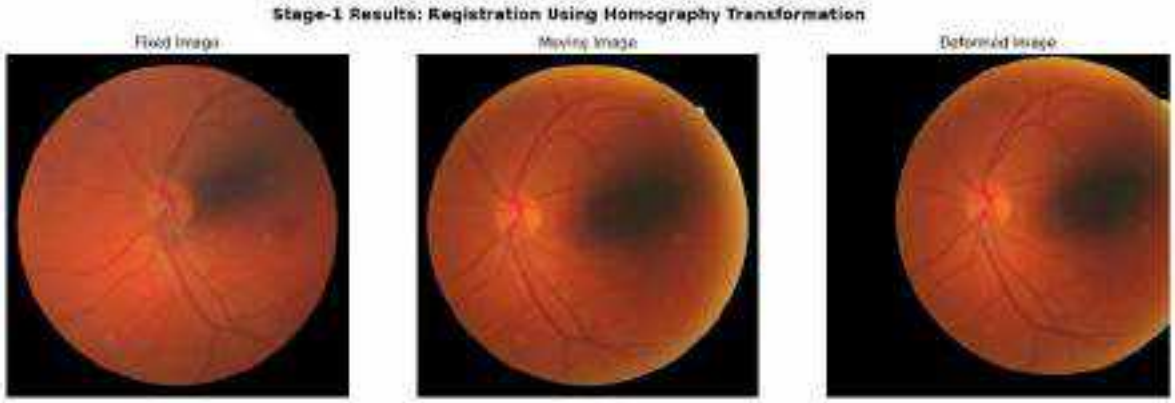
Note: 300 point correspondences were identified by the model for stage-2



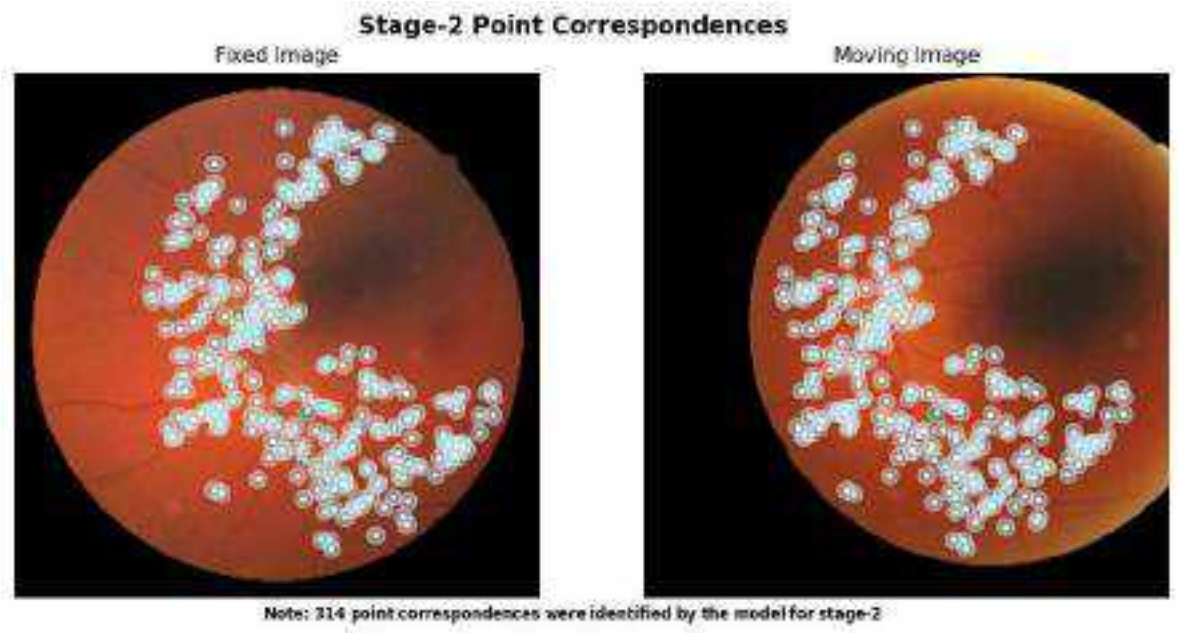
```
Mean Landmark Error for Case 45 Before Registration is 515.5014570513115 pixels
Mean Landmark Error for Case 45 After Registration is 3.3295900614376377 pixels
Case 46
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P47_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P47_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

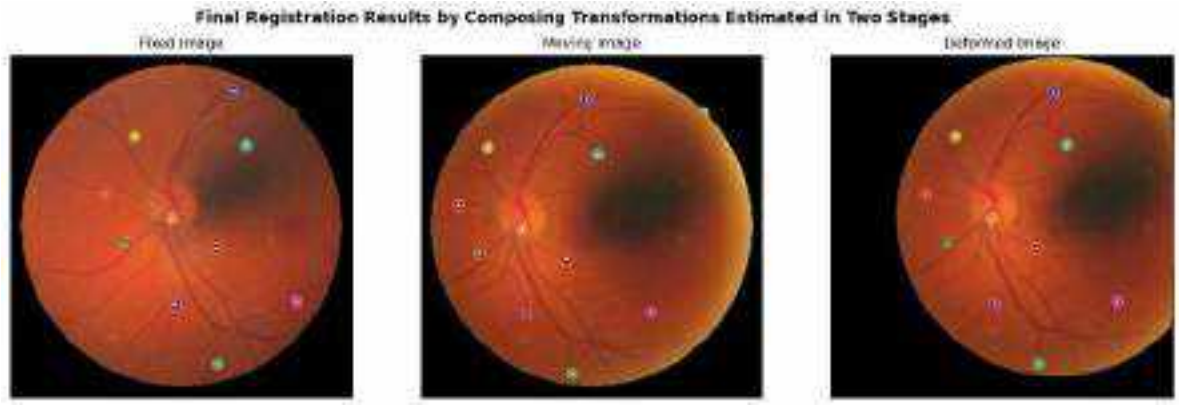



```
Homography Matrix:
[[ 1.03328599e+00 -1.28148214e-02  1.58207078e+02]
 [ 2.88894325e-02  1.00217765e+00 -3.48563138e+01]
 [ 5.67838889e-05 -1.88447510e-05  1.00000000e+00]]
```

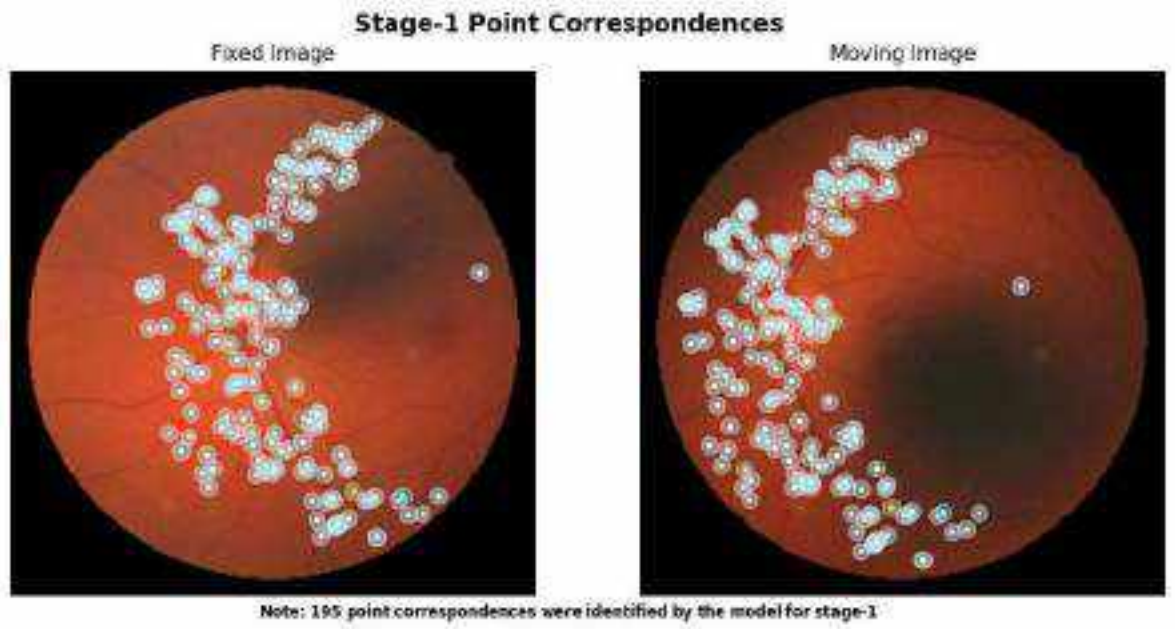



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





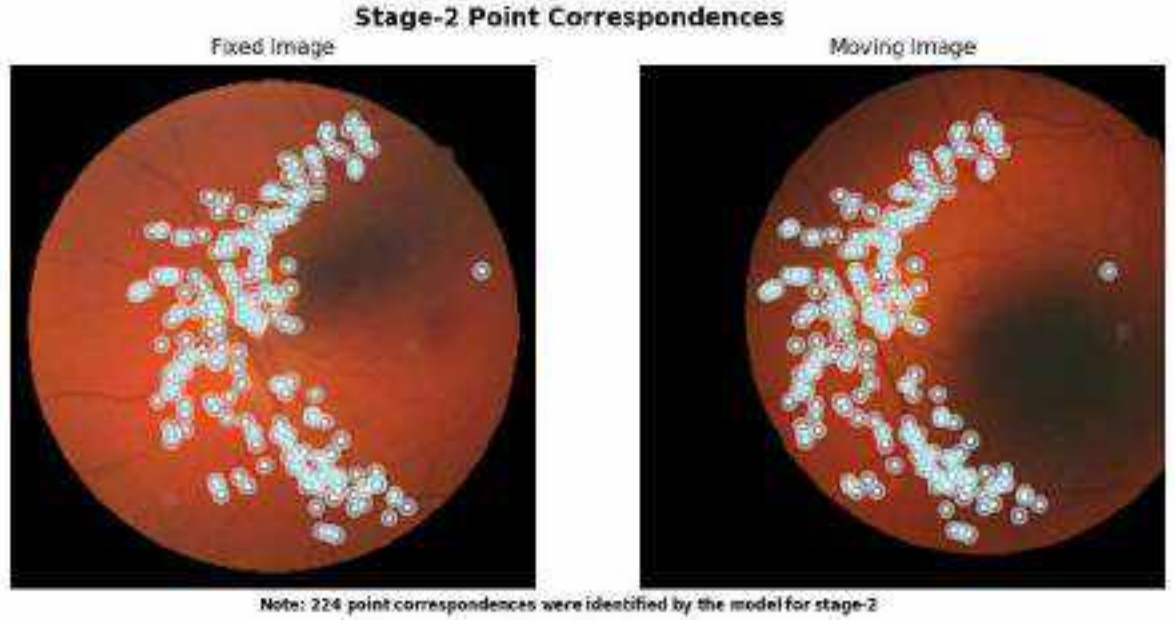

```
Mean Landmark Error for Case 46 Before Registration is 500.81101341234425 pixels
Mean Landmark Error for Case 46 After Registration is 3.594278762983038б pixels
Case 47
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P48_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P48_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



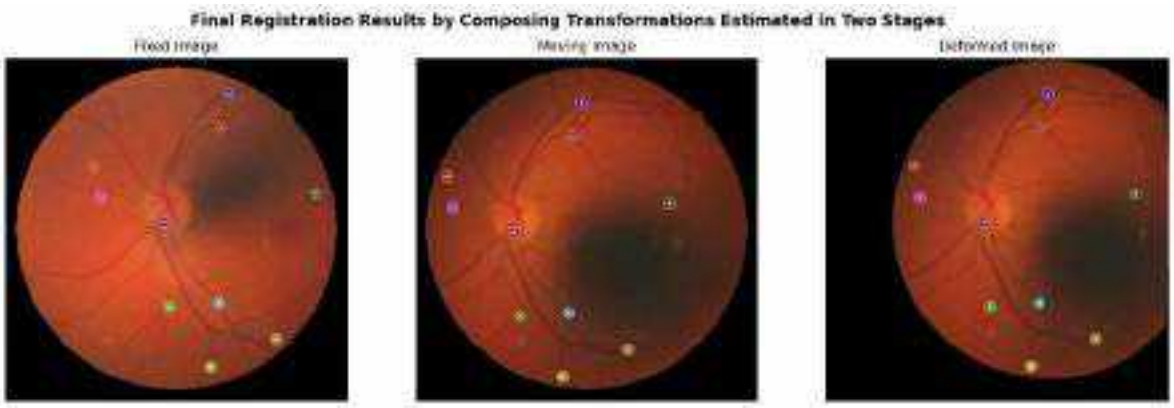
Note: 195 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.07310847e+00  4.66432461e-03  1.51273416e+02]
 [ 3.97295067e-02  1.03694330e+00 -4.16063063e+01]
 [ 9.91254758e-05  2.37720791e-06  1.00000000e+00]]
```

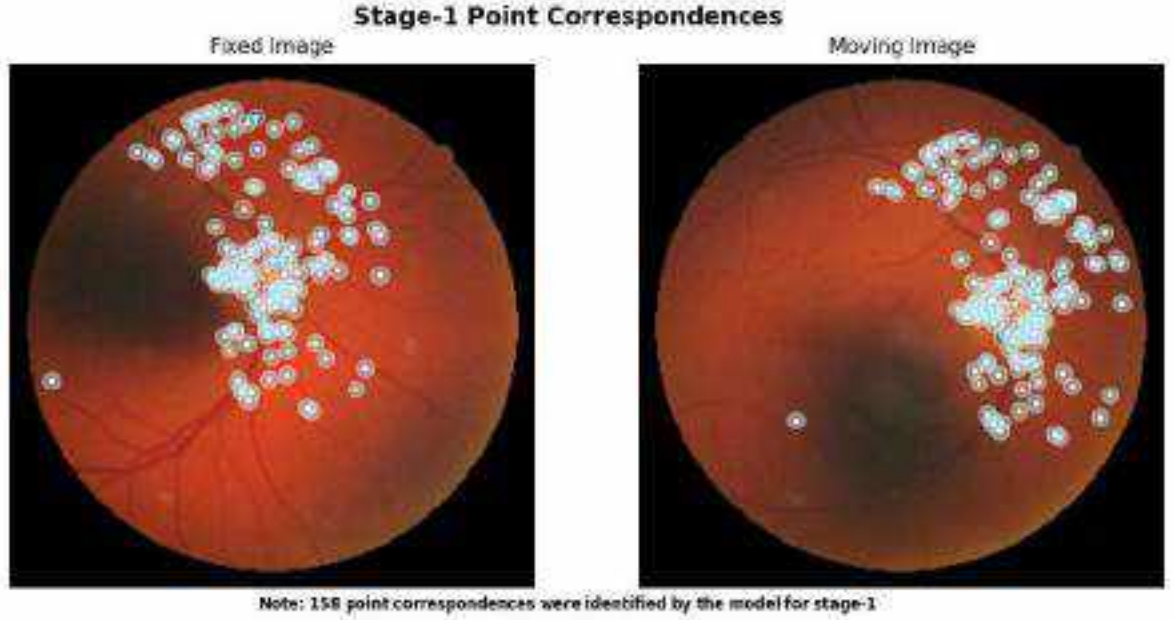

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 224 point correspondences were identified by the model for stage-2

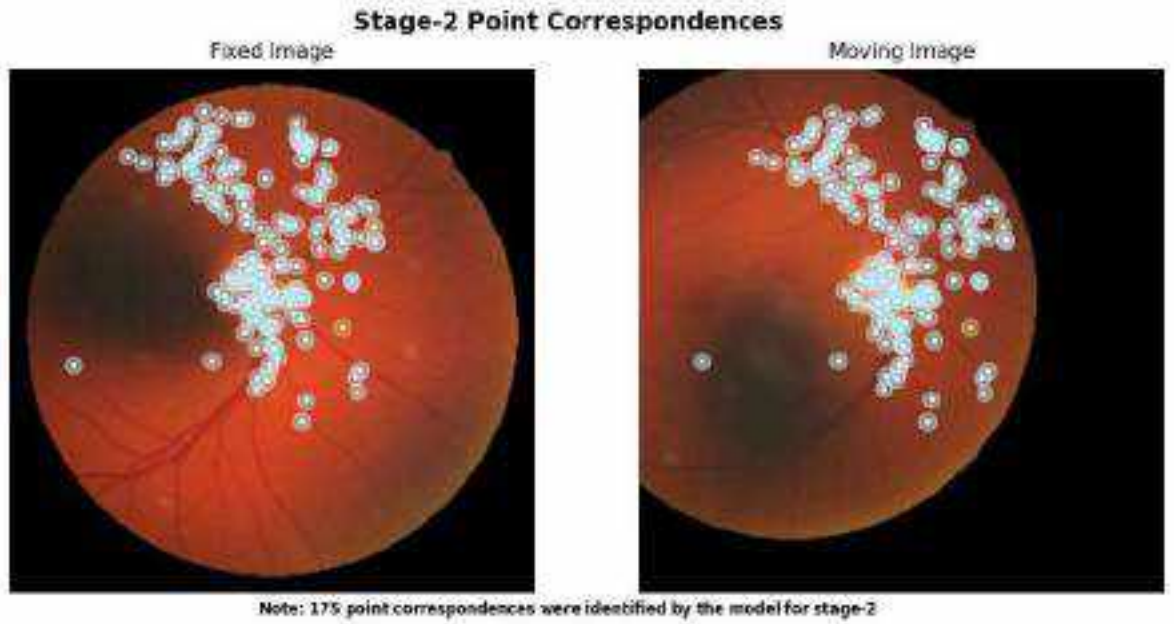



```
Mean Landmark Error for Case 47 Before Registration is 505.46033111106027 pixels
Mean Landmark Error for Case 47 After Registration is 4.058001630108471 pixels
Case 48
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/P49_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/P49_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

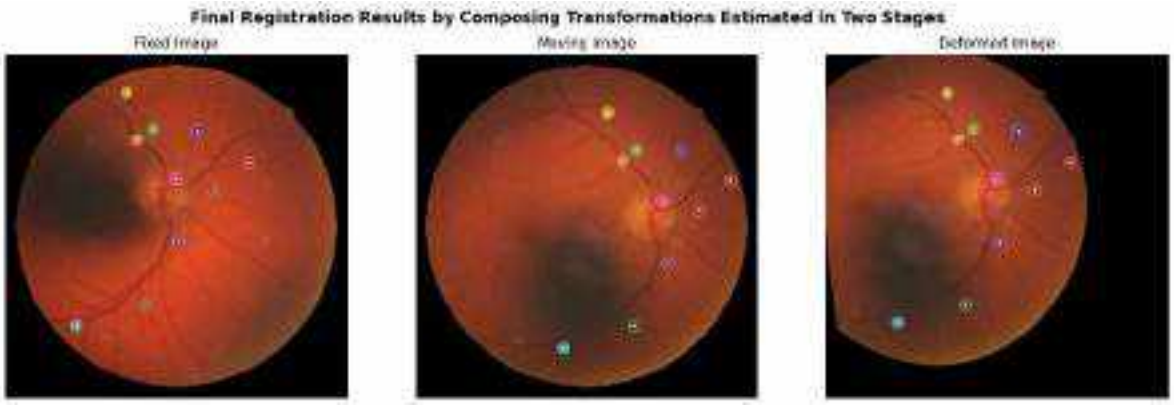



```
Homography Matrix:
[[ 8.60981011e-01 -6.85955713e-02 -1.33088010e+02]
 [-1.28472915e-02  8.79272432e-01 -3.55436713e+01]
 [-1.29648363e-04 -4.55592915e-05  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

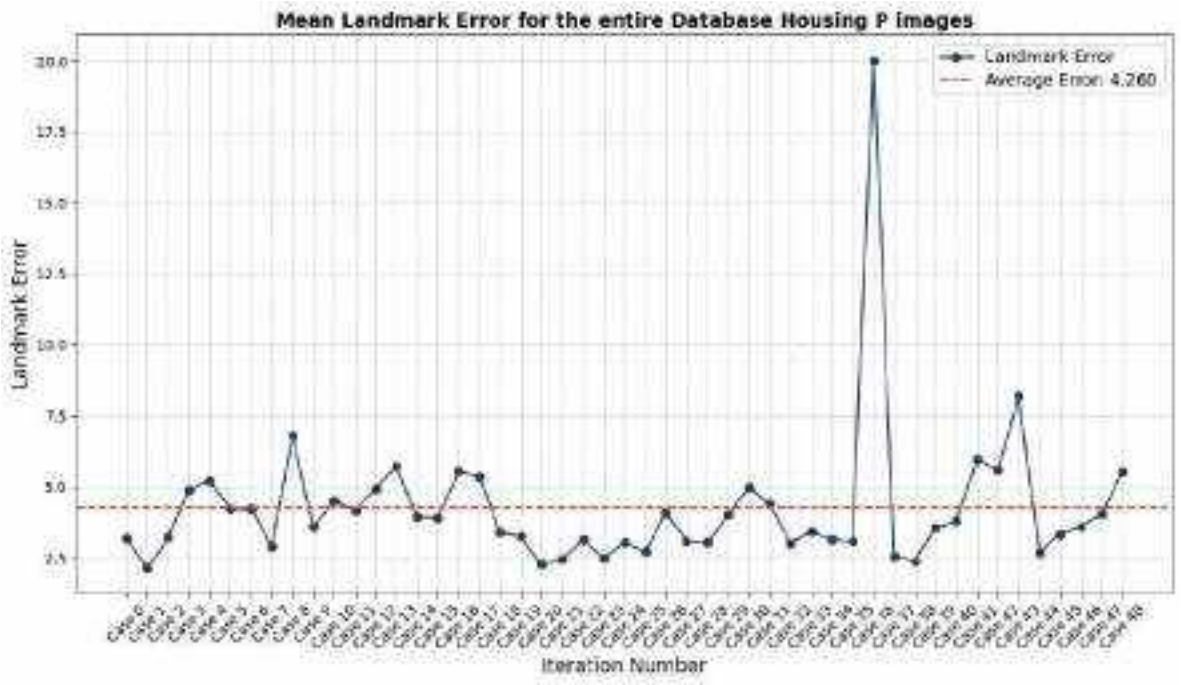


Note: 175 point correspondences were identified by the model for stage-2

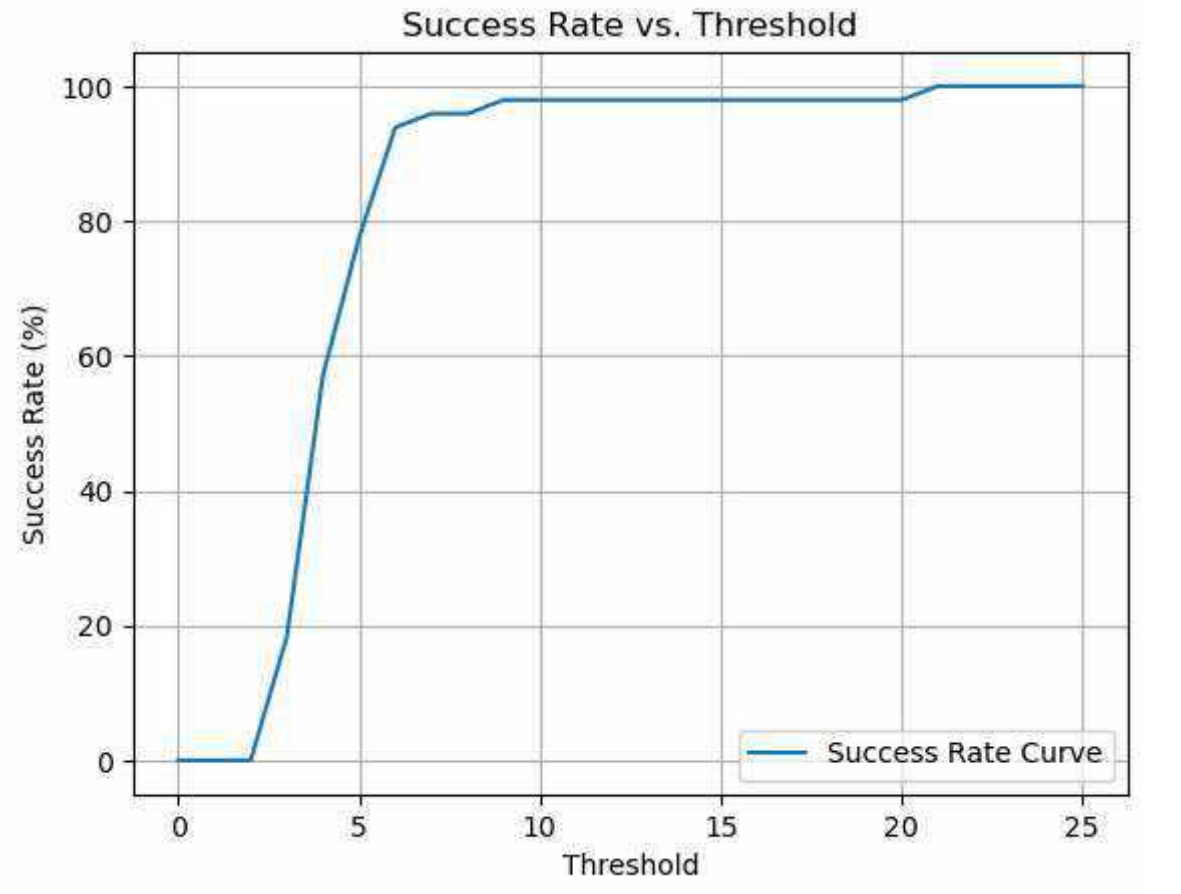



```
Mean Landmark Error for Case 48 Before Registration is 661.9757943843924 pixels
Mean Landmark Error for Case 48 After Registration is 5.521730422945O5 pixels
```

In [23]:
```
plot_landmark_errors(landmark_errors2,os.path.join(os.getcwd(),'FIRE_Image_Regis
```



In [24]:
```
compute_plot_FIRE_AUC(landmark_errors2)
```



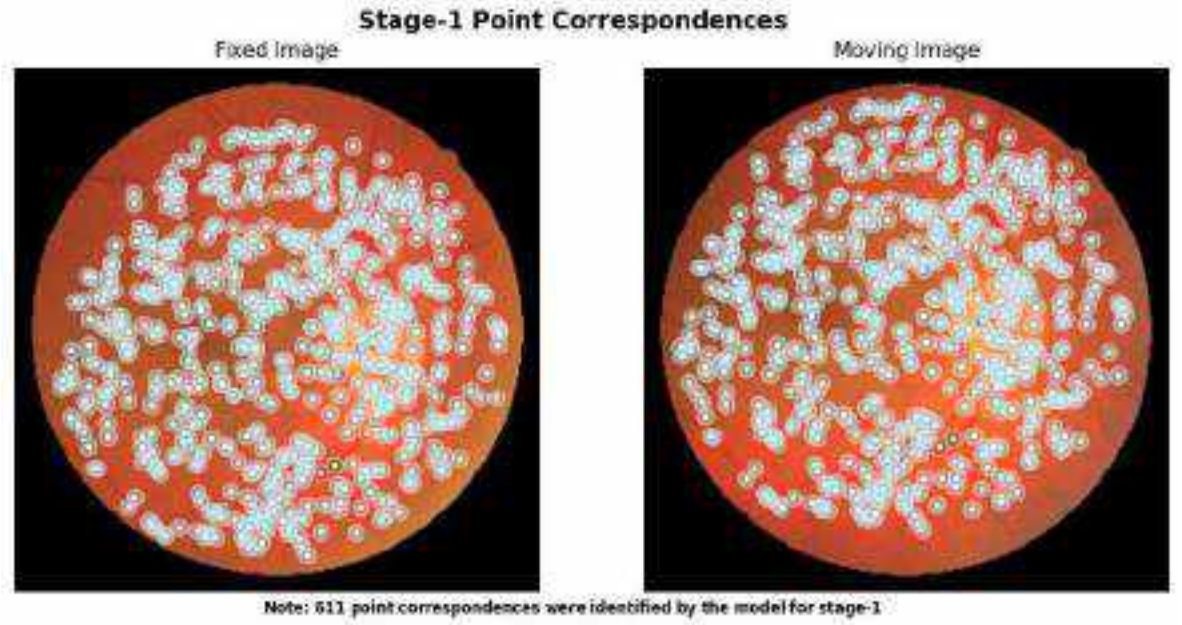

```
AUC: 0.8457142857142856
```

Class-S

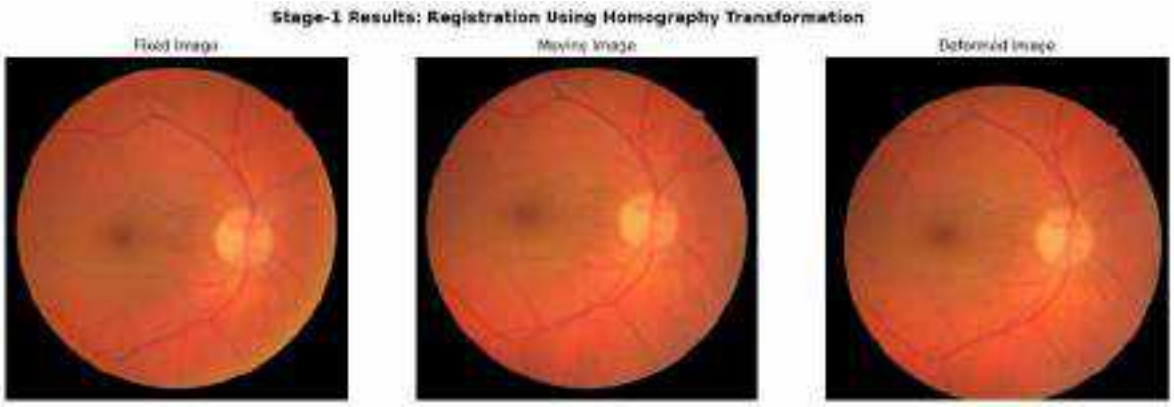
In [26]:
```
landmark_errors3=[]
for i in range(len(images_S)):
    print("Case {}".format(i))
    print("Loading Fixed Images {0} Moving Image{1} to the framework".format(ima
    original_low_res,computed_low_res = main(images_S[i],os.path.join(os.getcwd(
    imags,imgs,homography_matrix_low_res = compute_homography_matrix_and_plot(im
```

```
    if len(homography_matrix_low_res) !=0:
        transformed_points_hom = transform_points_homography(scaled_moving_point
        transformed_points_high_res_hom =  coordinates_rescaling(transformed_poi
        original_low_res,computed_low_res = main(imags,os.path.join(os.getcwd(),
        imag,imgs,polynomial_matrix_low_res = compute_third_order_polynomial_mat
        if len(polynomial_matrix_low_res) !=0:
            ## rescaled version for dispaly purposes
            transformed_points_poly = transform_points_third_order_polynomial(tr
            original_image_point_correspondences(imag,images_S[i][0],img_size, s
            ### Original Version for computation of errors
            polynomial_matrix = transform_points_third_order_polynomial_matrix(c
            bef_error = compute_landmark_error(fixed_points_S[i],fixed_image_siz
            aft_error = compute_landmark_error_fixed_space(polynomial_matrix,fix
            print("Mean Landmark Error for Case {0} Before Registration is {1} p
            print("Mean Landmark Error for Case {0} After Registration is {1} pi
            landmark_errors3.append(aft_error)
        else:
            landmark_errors3.append(10000)
    else:
        landmark_errors3.append(10000)
```

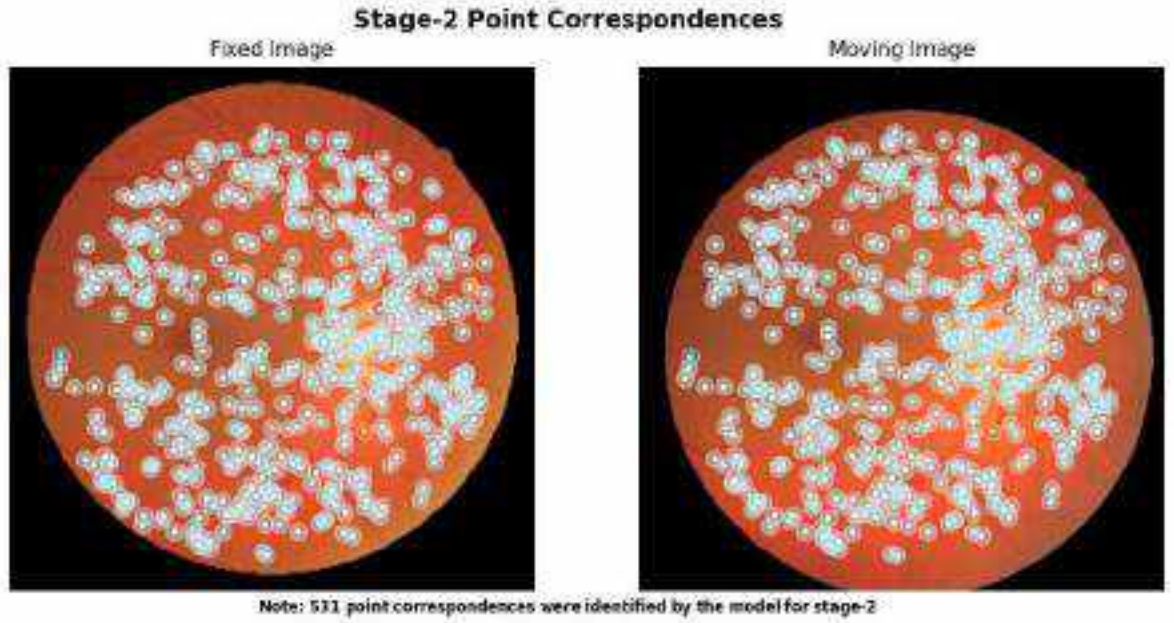
```
Case 0
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S01_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S01_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
/scratch/local/29752099/ipykernel_2239179/635223464.py:100: UserWarning: To copy
construct from a tensor, it is recommended to use sourceTensor.clone().detach() o
r sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(so
urceTensor).
  points_indices = torch.tensor(pts_list)
```



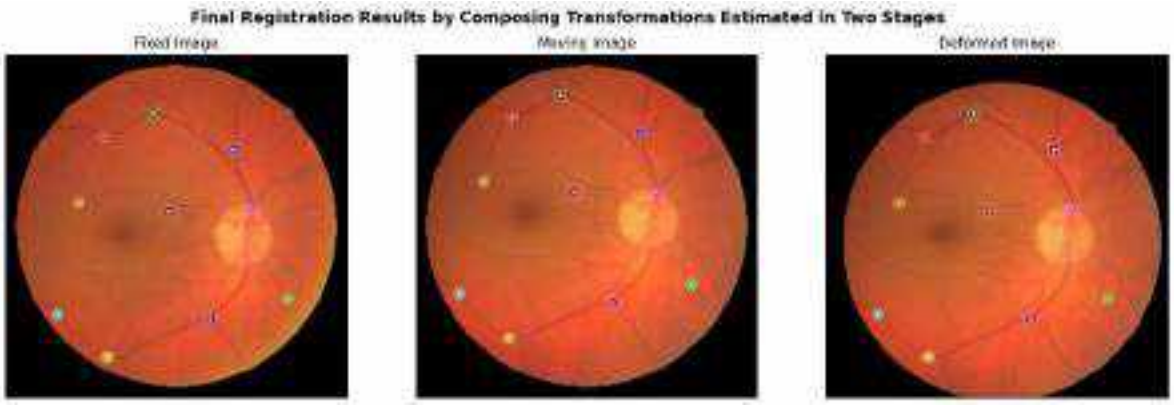

```
Homography Matrix:
[[ 1.01597681e+00  3.99915377e-02 -1.32503826e+00]
 [-2.31004417e-02  1.02474530e+00  5.74226344e+01]
 [ 9.78259456e-06  2.88632029e-05  1.00000000e+00]]
```



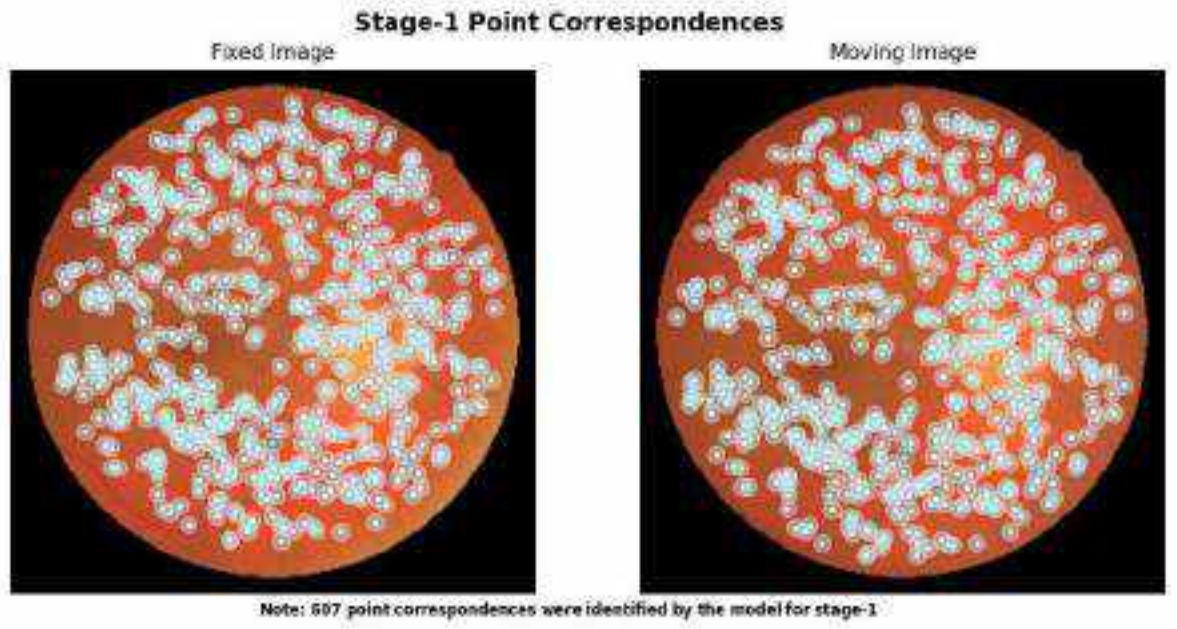

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 531 point correspondences were identified by the model for stage-2

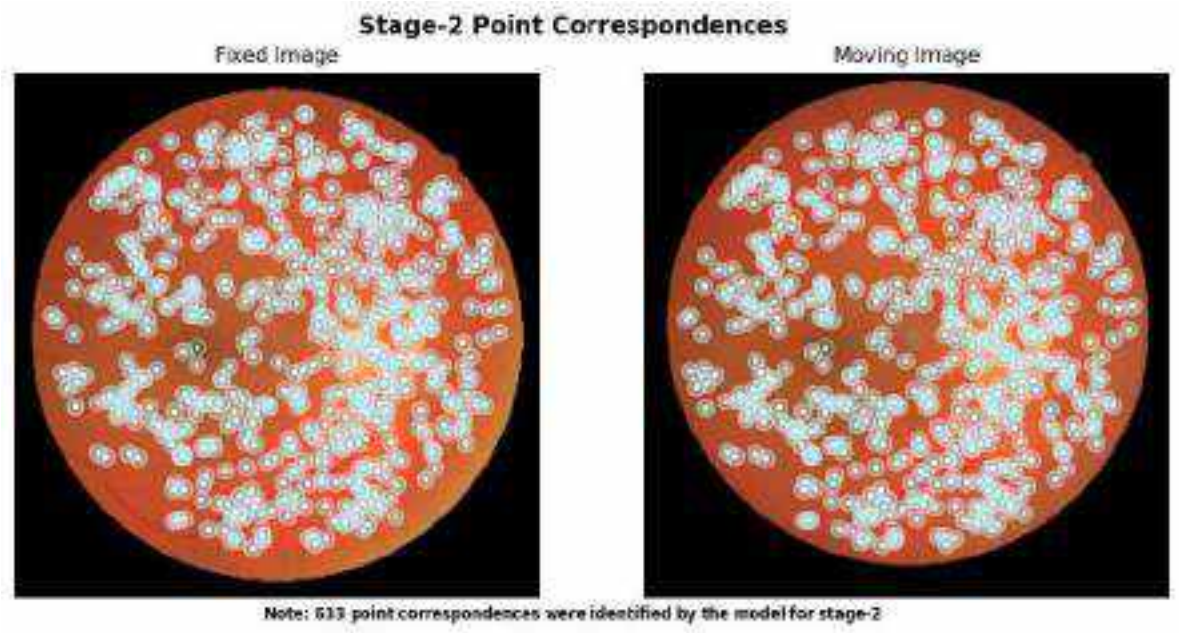


```
Mean Landmark Error for Case 0 Before Registration is 164.3500949592529 pixels
Mean Landmark Error for Case 0 After Registration is 1.6376009327563996 pixels
Case 1
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S02_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S02_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

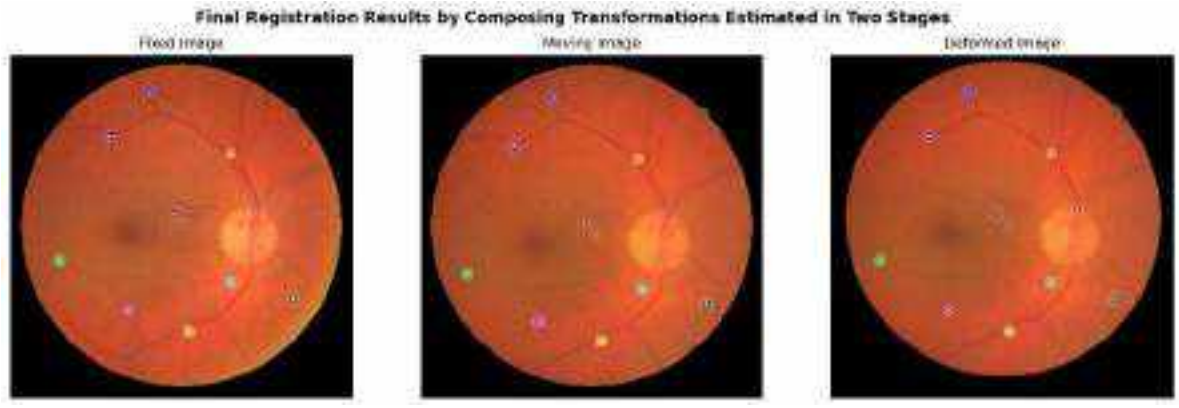



```
Homography Matrix:
[[ 9.81934074e-01 -3.82160250e-02  2.86591344e+01]
 [ 3.60782793e-02  9.74159339e-01 -2.77176722e+01]
 [ 4.21596850e-06 -9.69023653e-06  1.00000000e+00]]
```

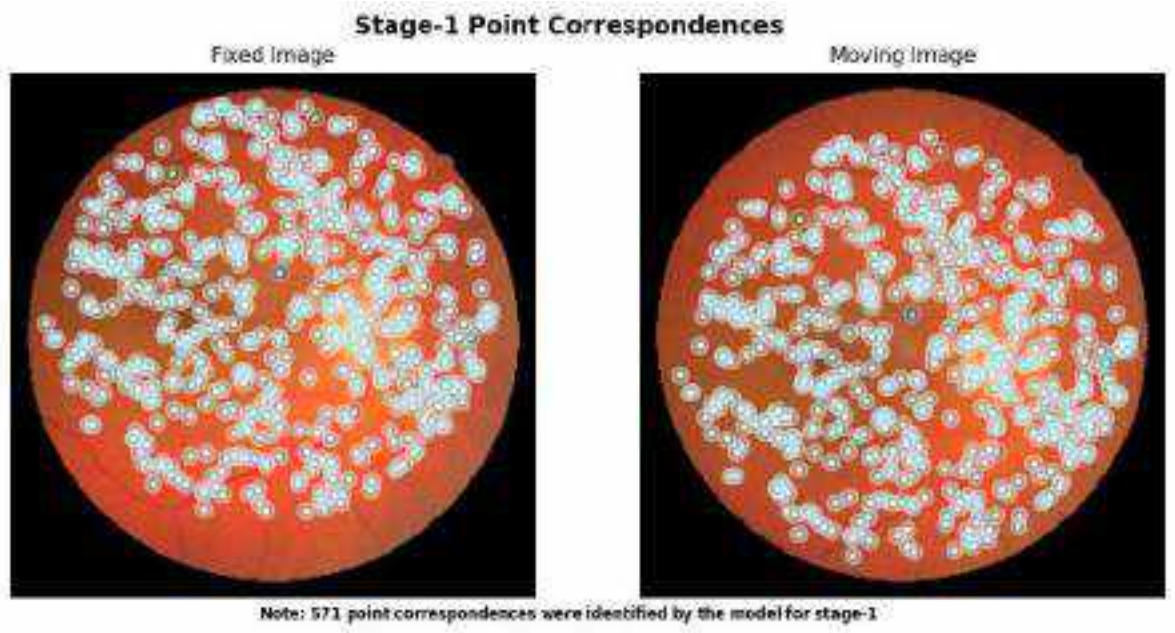

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 513 point correspondences were identified by the model for stage-2

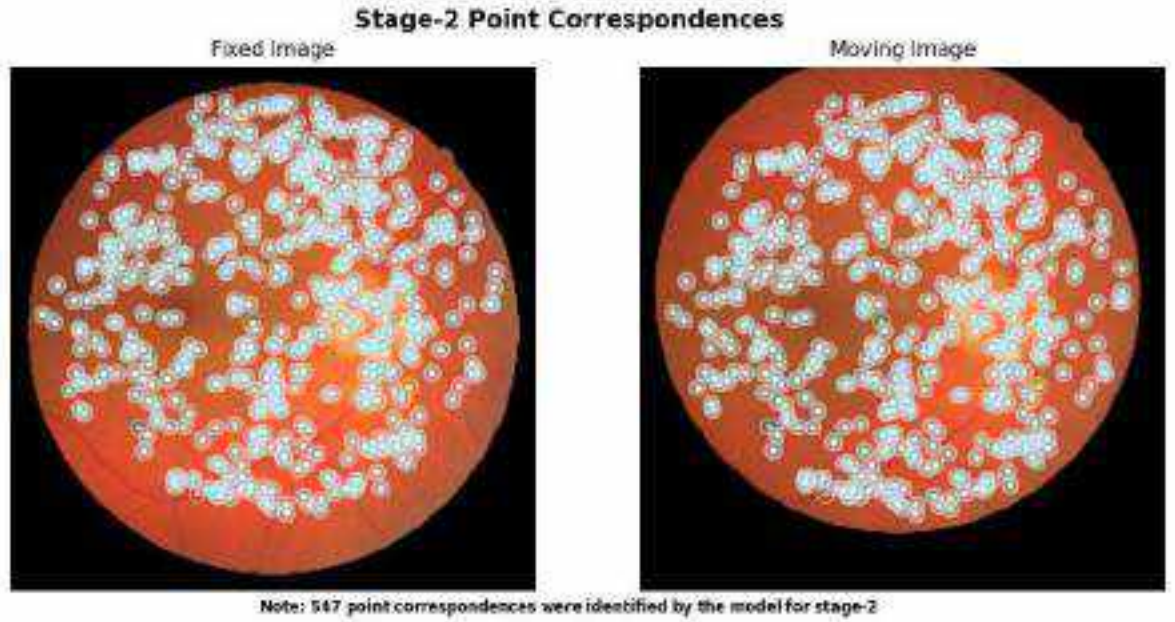


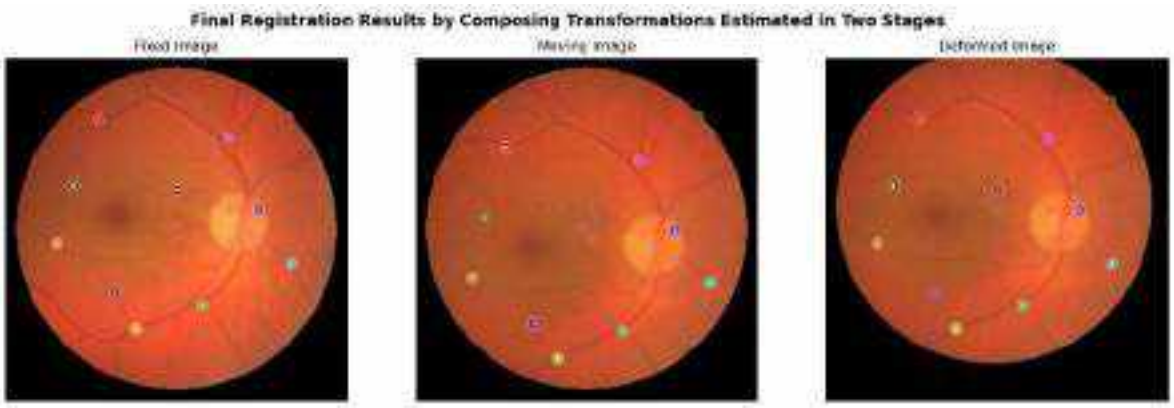

```
Mean Landmark Error for Case 1 Before Registration is 75.99915733644708 pixels
Mean Landmark Error for Case 1 After Registration is 1.4785059071762228 pixels
Case 2
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S03_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S03_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



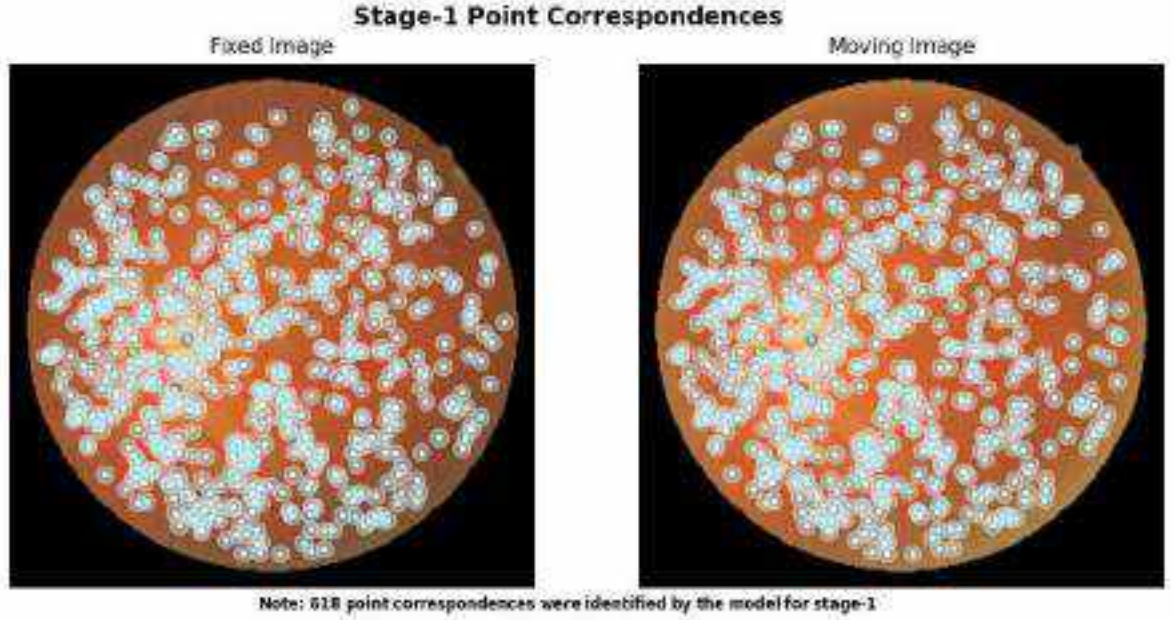

```
Homography Matrix:
[[ 9.58127471e-01 -7.51058250e-02  3.36965019e+01]
 [ 5.58427666e-02  9.46647361e-01 -8.11314609e+01]
 [-1.24998249e-05 -3.46829024e-05  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



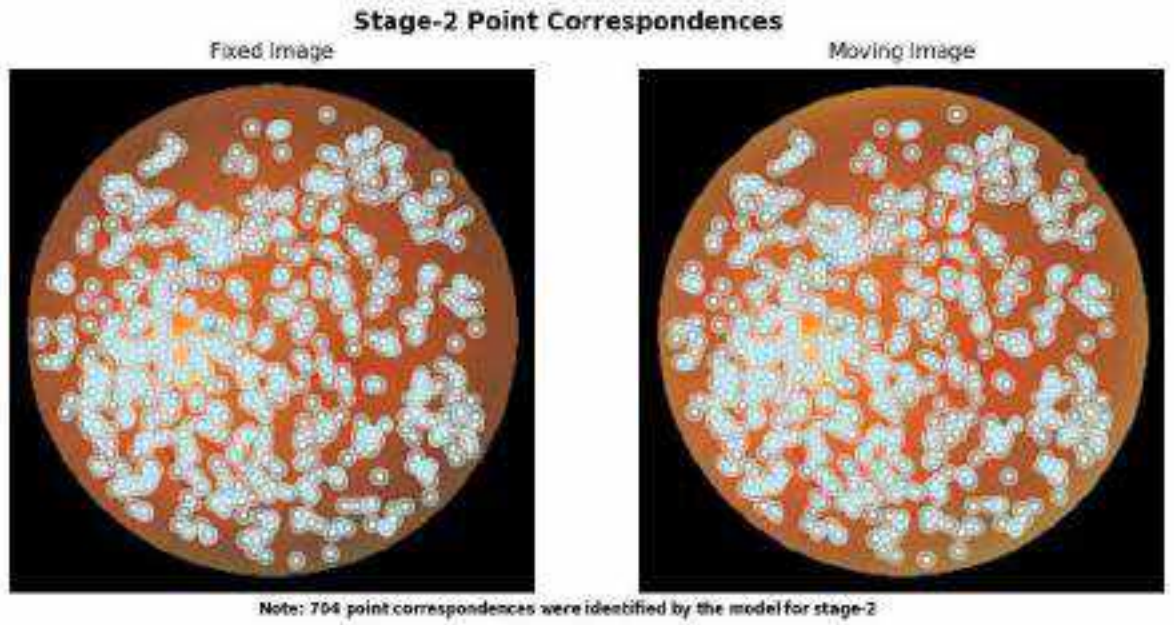



```
Mean Landmark Error for Case 2 Before Registration is 237.37049636940247 pixels
Mean Landmark Error for Case 2 After Registration is 2.2332959405416615 pixels
Case 3
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S04_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S04_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

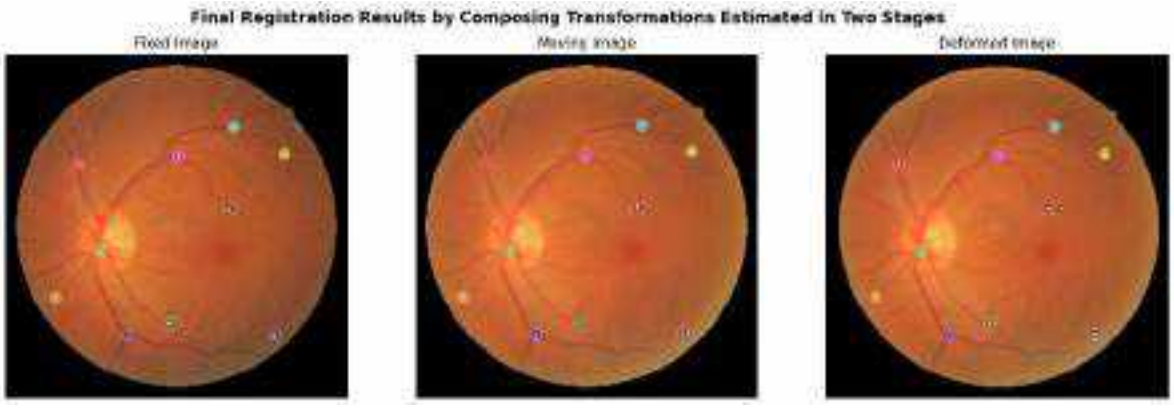



```
Homography Matrix:
[[ 1.01339749e+00 -1.20355312e-03  4.22441472e+00]
 [ 1.16629946e-02  1.01219367e+00 -3.46772364e+00]
 [ 8.54217294e-06  1.13163583e-05  1.00000000e+00]]
```

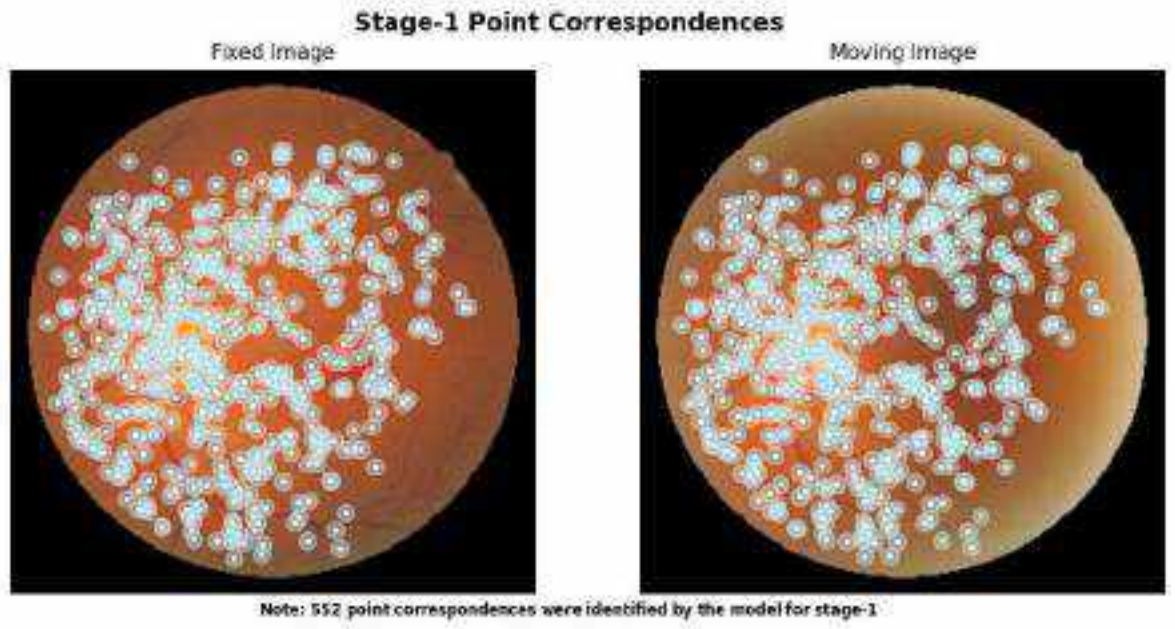

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



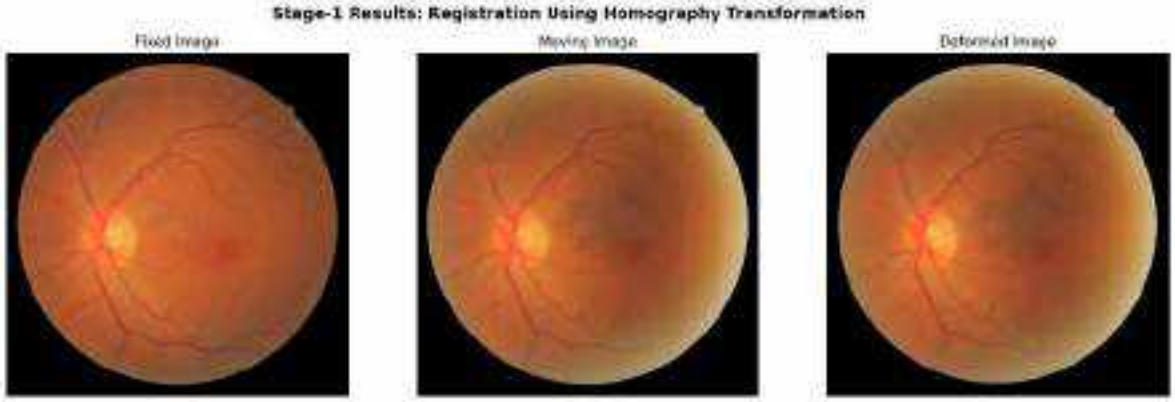
Note: 704 point correspondences were identified by the model for stage-2



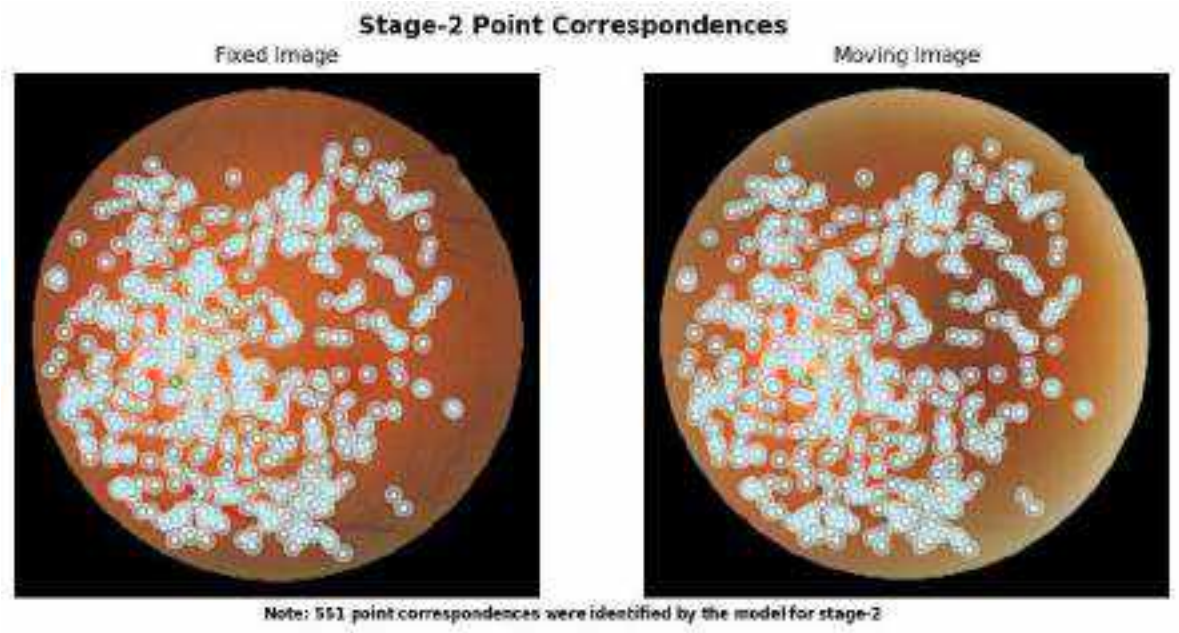
```
Mean Landmark Error for Case 3 Before Registration is 21.640081738168478 pixels
Mean Landmark Error for Case 3 After Registration is 2.0328717602794777 pixels
Case 4
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S05_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S05_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

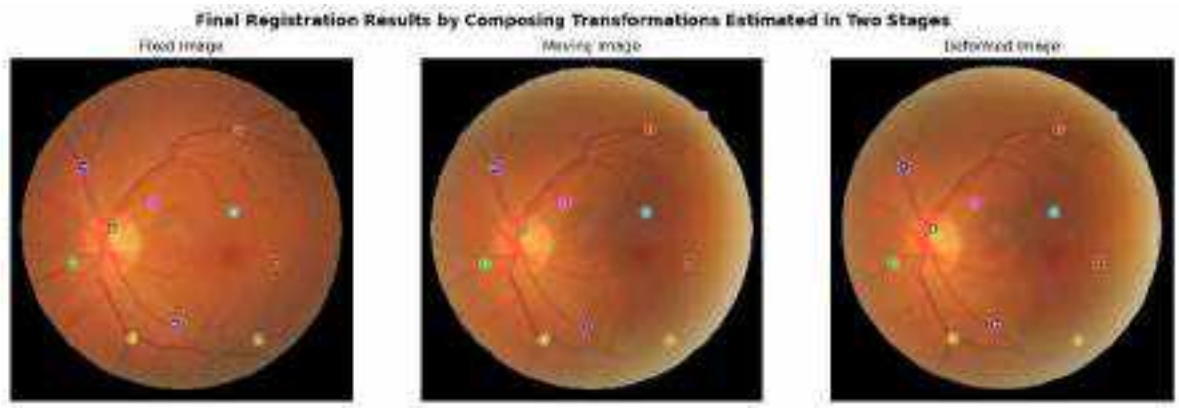



```
Homography Matrix:
[[ 9.97720285e-01 -3.80512149e-03  3.65657831e-01]
 [ 3.54803743e-03  9.99628007e-01 -2.02633141e+00]
 [-8.14505742e-07  3.05834764e-06  1.00000000e+00]]
```

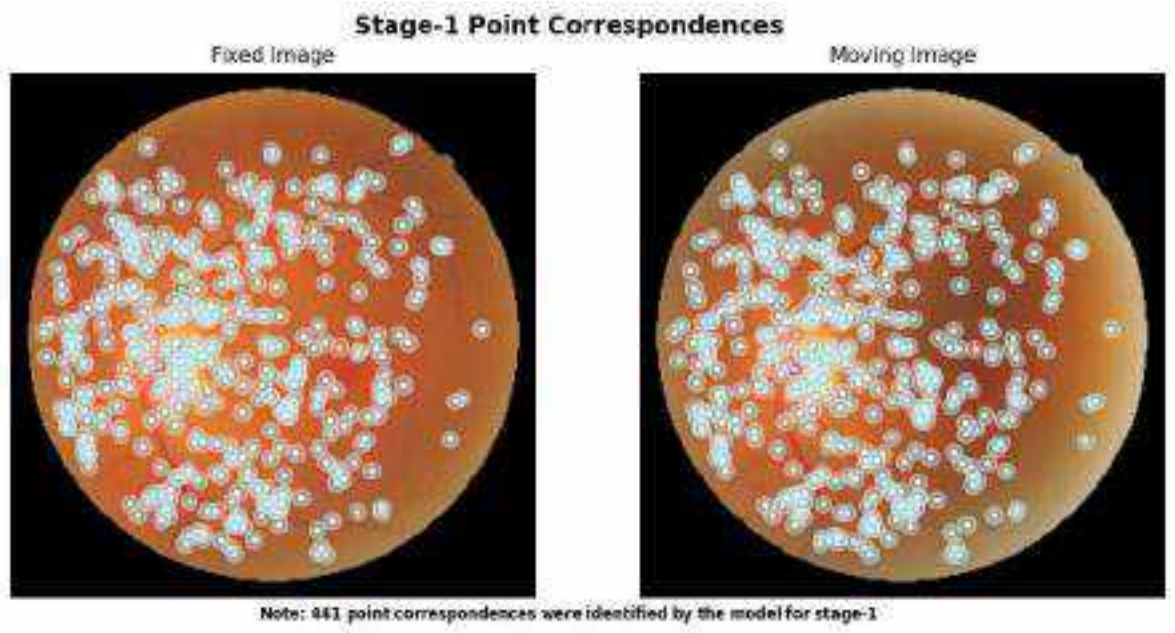



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





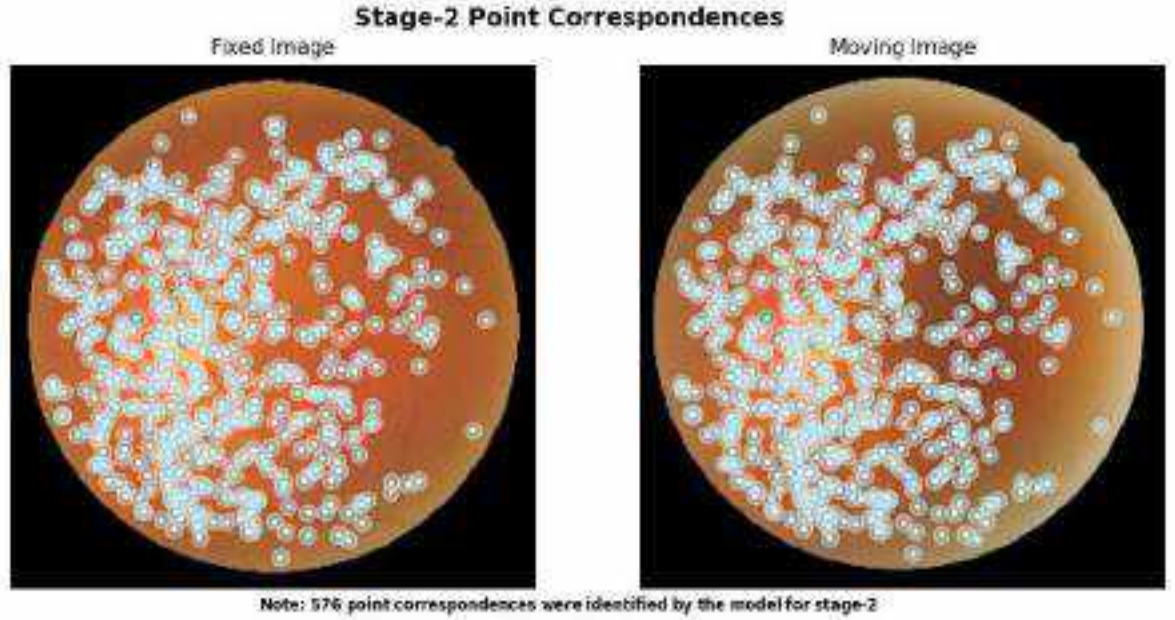

```
Mean Landmark Error for Case 4 Before Registration is 12.636911772093137 pixels
Mean Landmark Error for Case 4 After Registration is 1.2554157300396682 pixels
Case 5
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S06_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S06_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



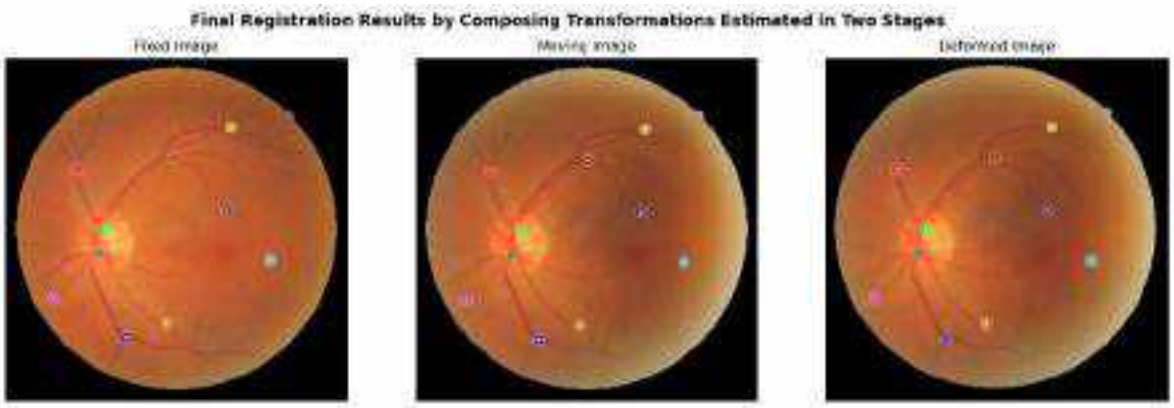
Note: 441 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.87848576e-01  5.73058534e-04 -6.20258456e+00]
 [-6.49265239e-03  9.95791225e-01 -2.28377619e+00]
 [-1.29285009e-05  4.53240024e-07  1.00000000e+00]]
```

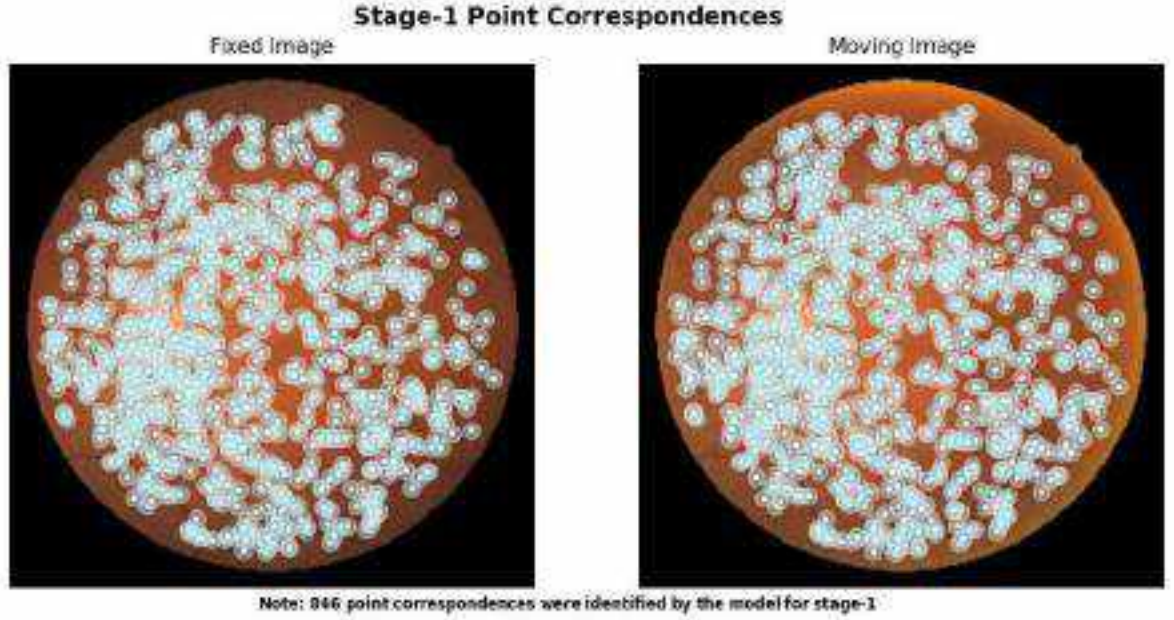

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



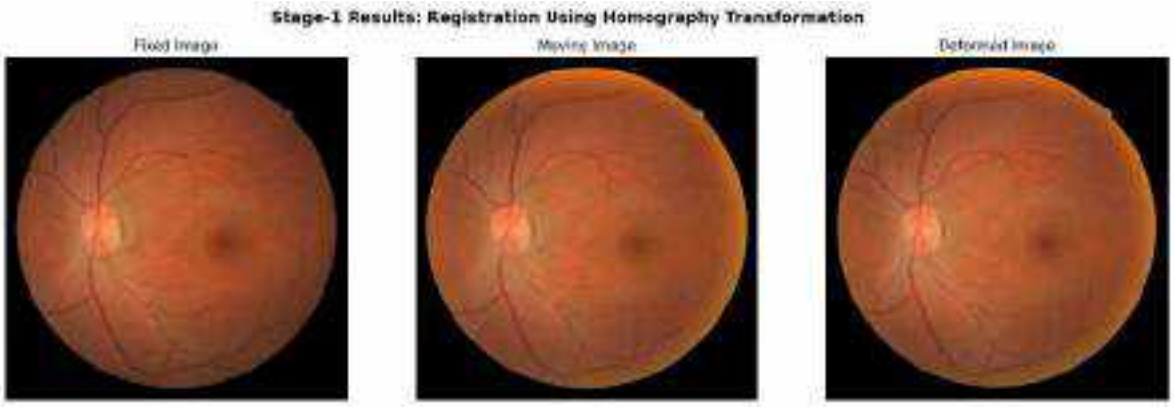
Note: 576 point correspondences were identified by the model for stage-2



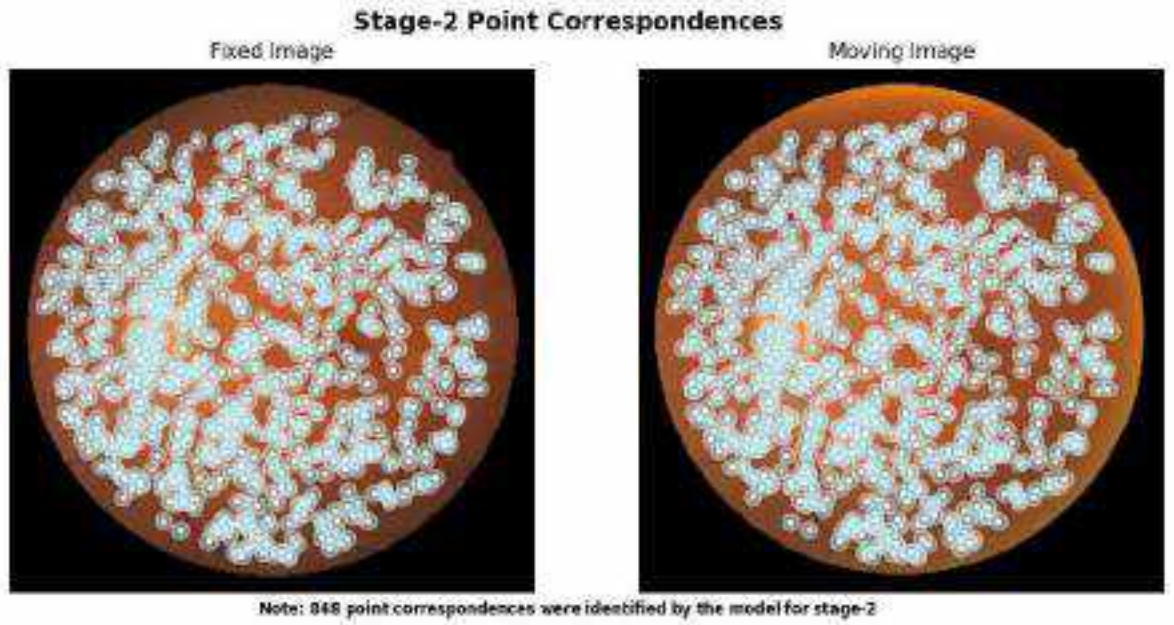

```
Mean Landmark Error for Case 5 Before Registration is 32.65258800622144 pixels
Mean Landmark Error for Case 5 After Registration is 1.9559382181676237 pixels
Case 6
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S07_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S07_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



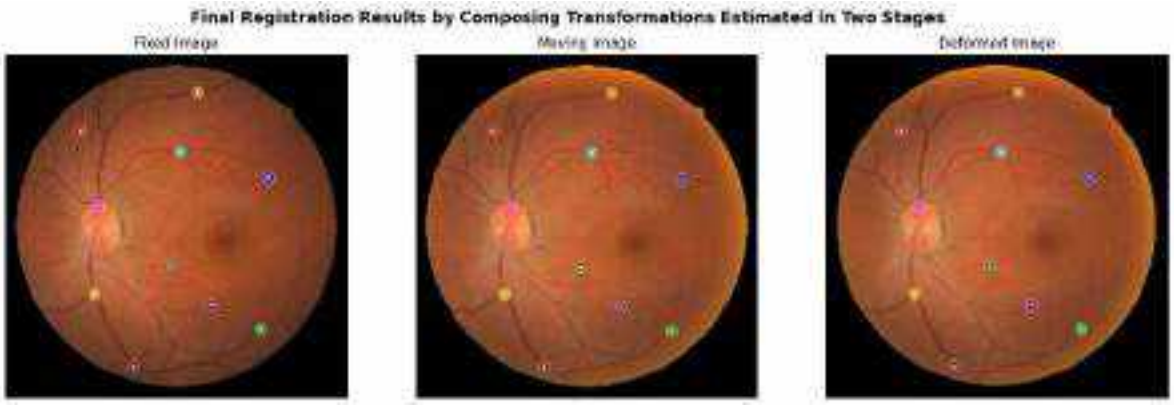

```
Homography Matrix:
[[ 9.96946219e-01  9.45469131e-03 -5.56649244e+00]
 [-9.47992712e-03  9.97118426e-01  3.54893539e+00]
 [ 2.97783175e-07 -7.61284076e-07  1.00000000e+00]]
```



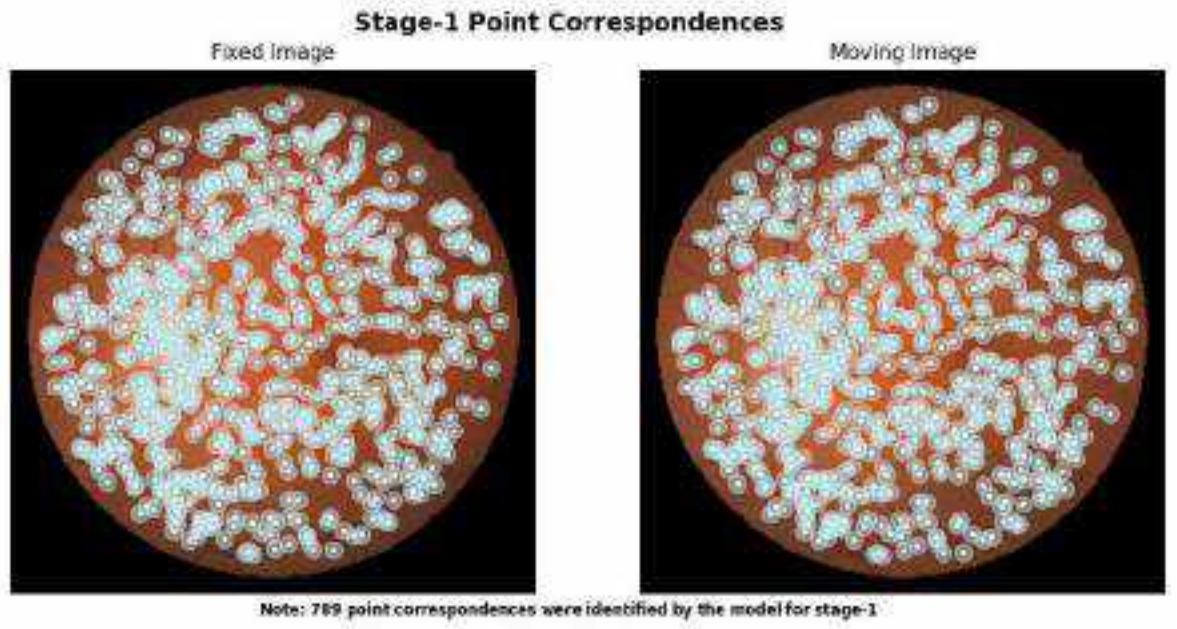

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



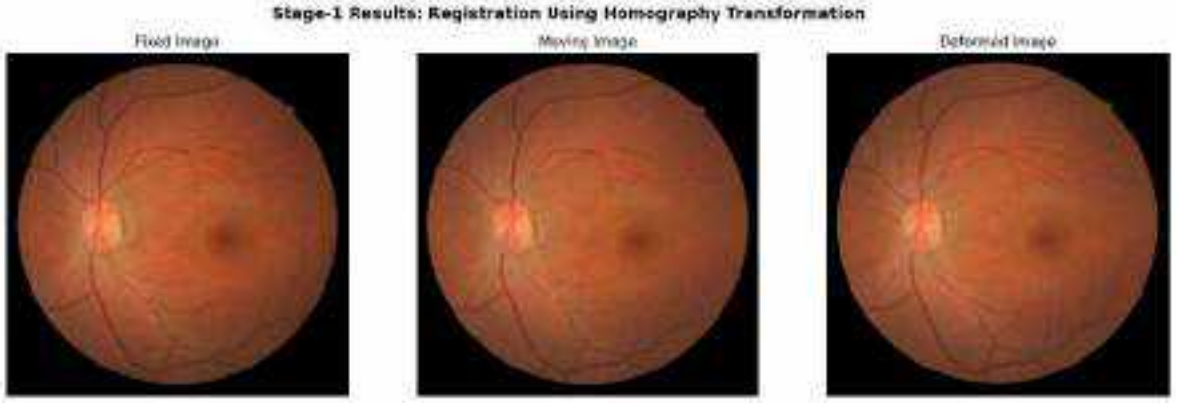
Note: 848 point correspondences were identified by the model for stage-2



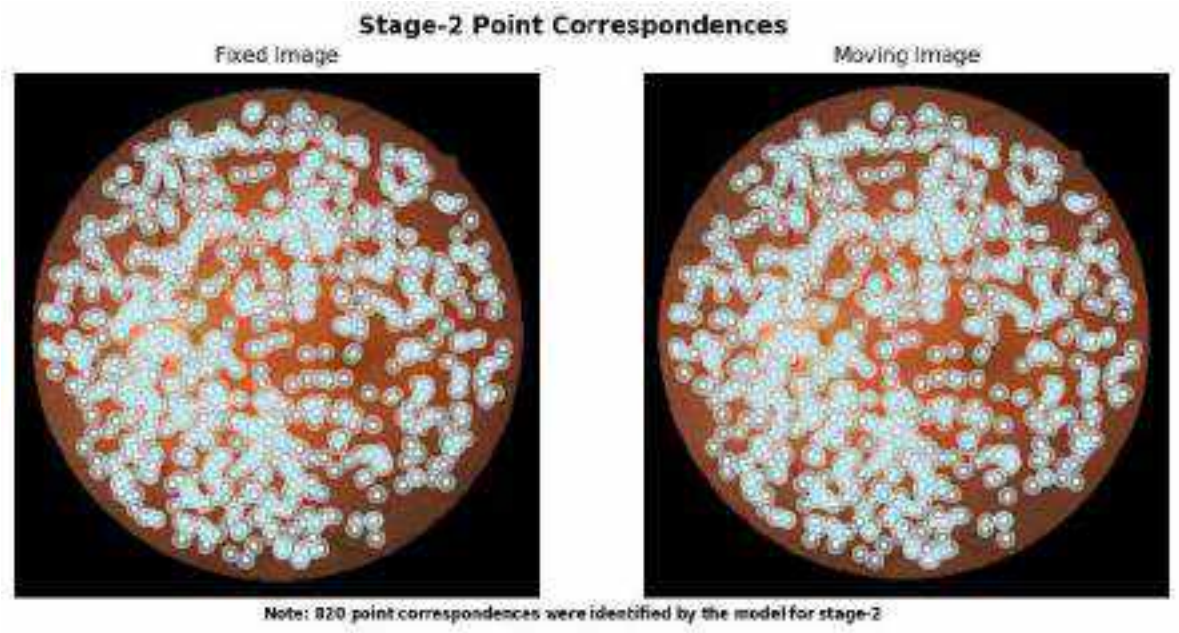
```
Mean Landmark Error for Case 6 Before Registration is 13.874744766406437 pixels
Mean Landmark Error for Case 6 After Registration is 1.0345796727758412 pixels
Case 7
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S08_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S08_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

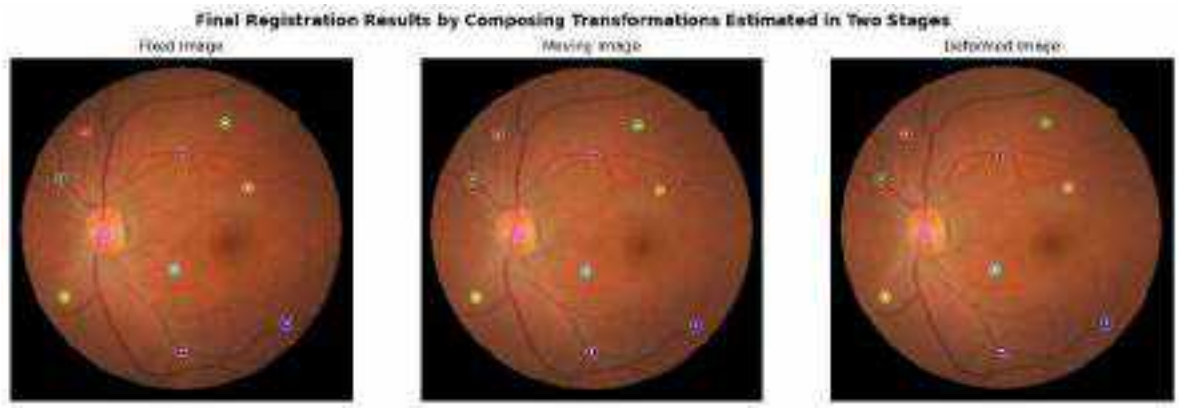



```
Homography Matrix:
[[ 9.97258622e-01  9.43694742e-03 -8.31758741e+00]
 [-1.26344452e-02  9.97105681e-01  1.13617286e+00]
 [-3.61329473e-06 -3.44321032e-06  1.00000000e+00]]
```

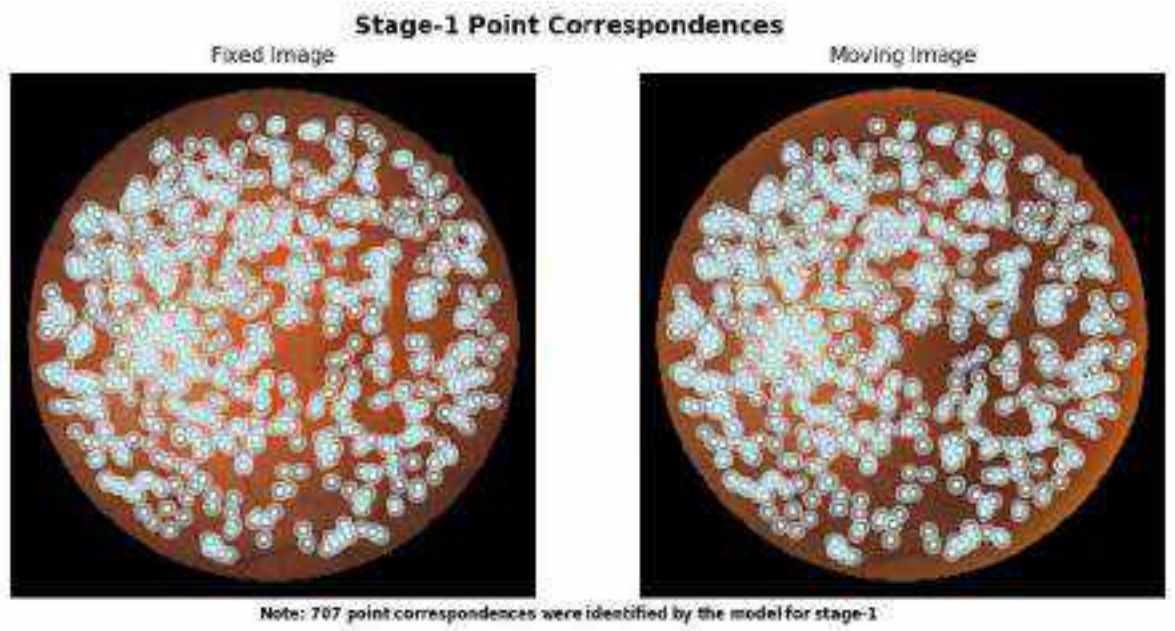



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



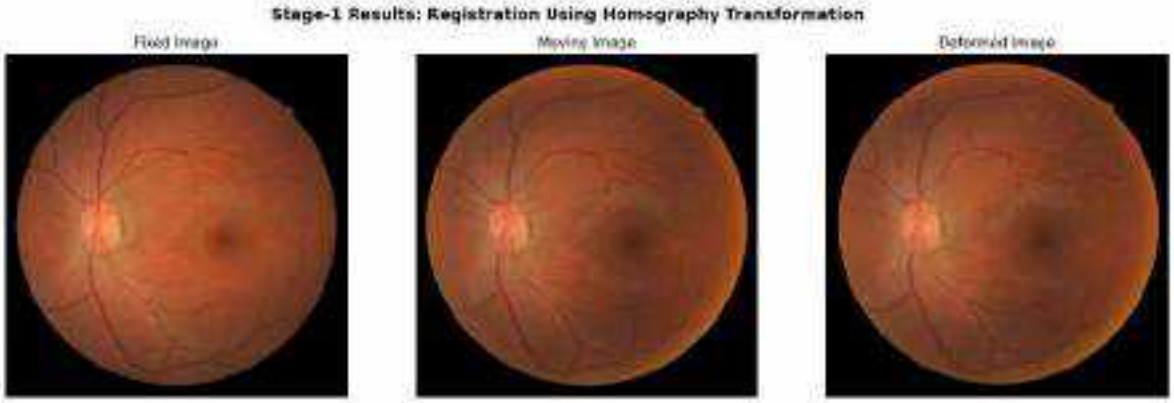



```
Mean Landmark Error for Case 7 Before Registration is 20.26197495652475 pixels
Mean Landmark Error for Case 7 After Registration is 0.9059810830828203 pixels
Case 8
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S09_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S09_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

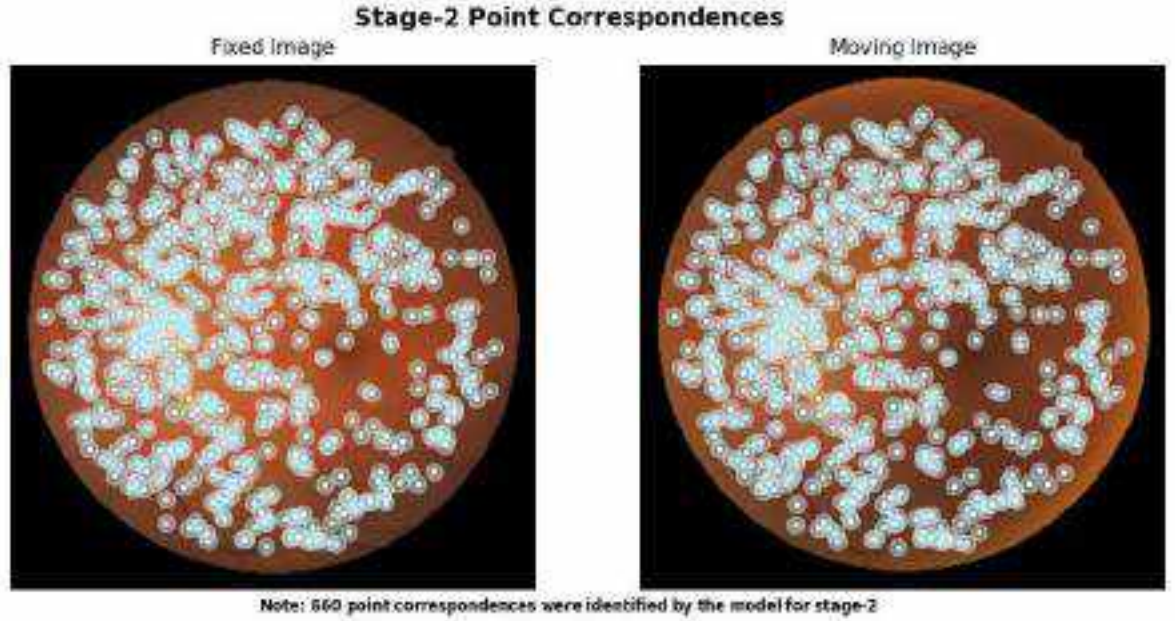


Note: 707 point correspondences were identified by the model for stage-1

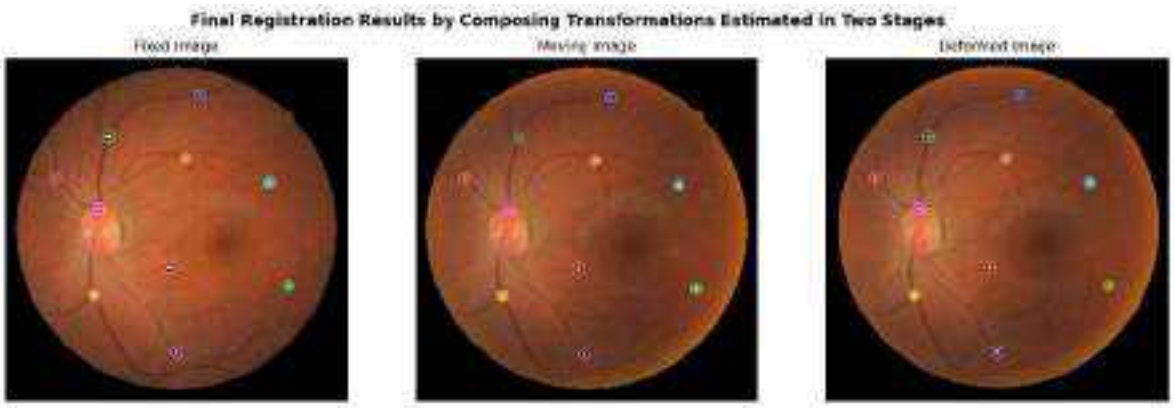
```
Homography Matrix:
[[ 1.00133399e+00  1.04407088e-02 -2.89835586e+00]
 [-1.33778133e-02  9.99193335e-01  7.95949727e-01]
 [ 7.20681778e-07 -4.00049682e-06  1.00000000e+00]]
```



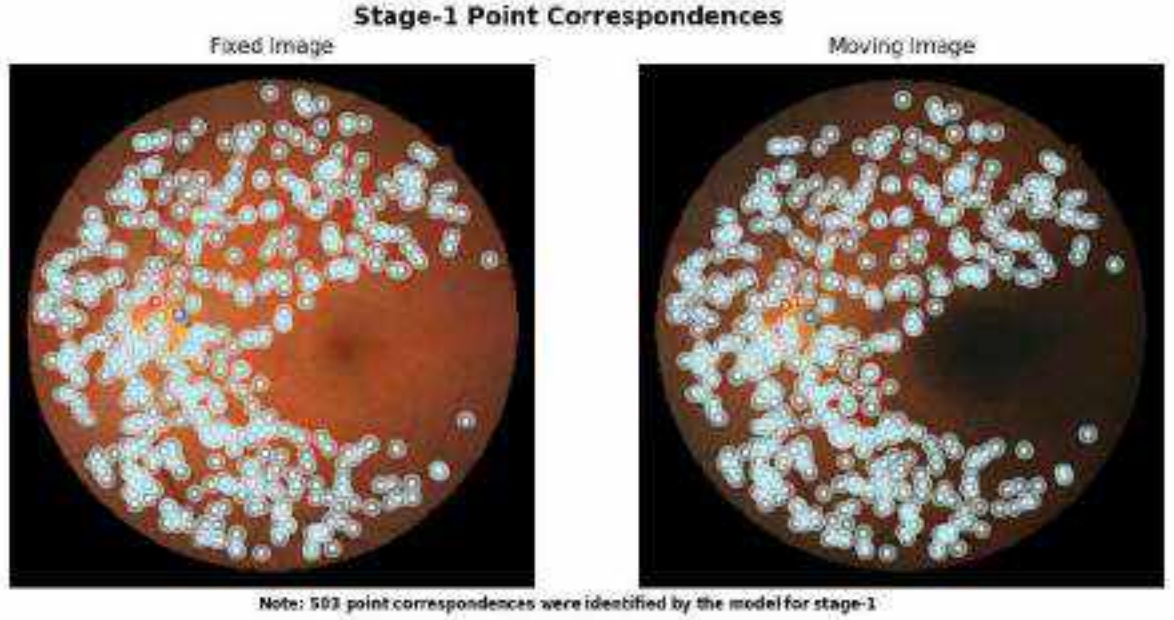

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



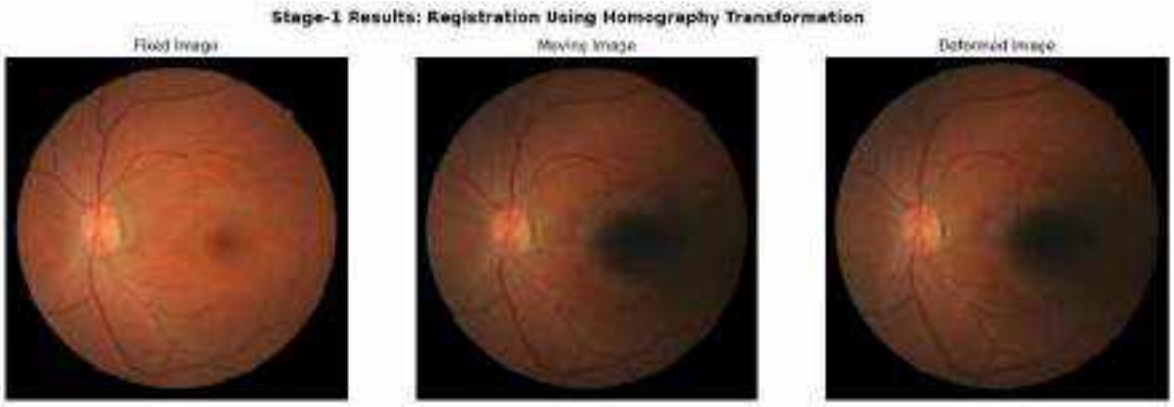
Note: 660 point correspondences were identified by the model for stage-2



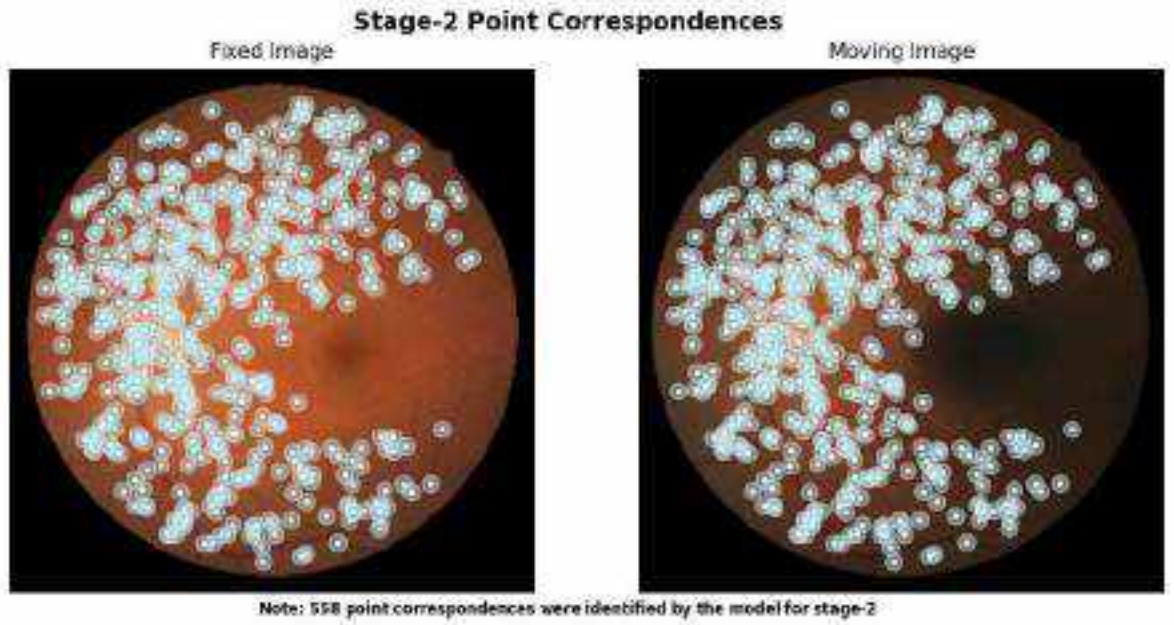

```
Mean Landmark Error for Case 8 Before Registration is 19.9050844076483398 pixels
Mean Landmark Error for Case 8 After Registration is 1.4825829415589253 pixels
Case 9
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S10_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S10_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



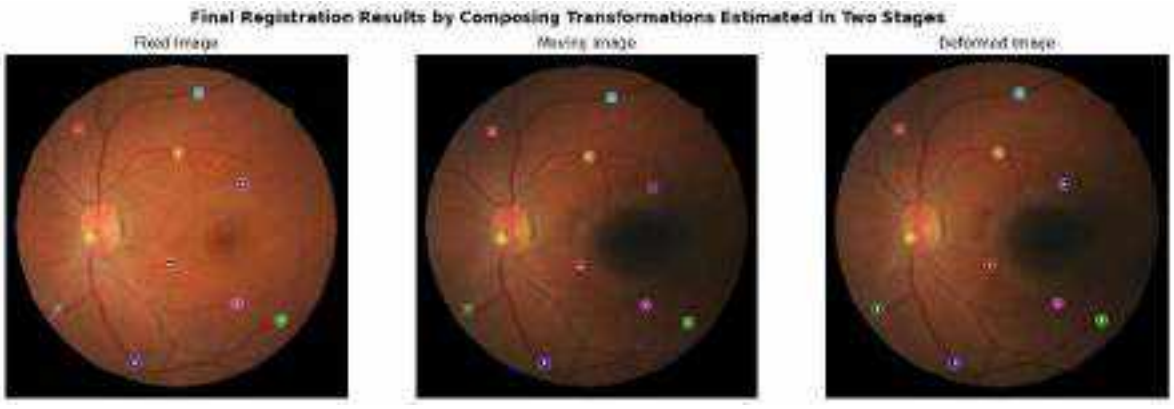

```
Homography Matrix:
[[ 1.01144914e+00  1.02540330e-02 -1.18094299e+01]
 [-1.57995470e-02  1.01064476e+00 -6.26379211e+00]
 [-7.89022194e-07 -6.30665703e-06  1.00000000e+00]]
```



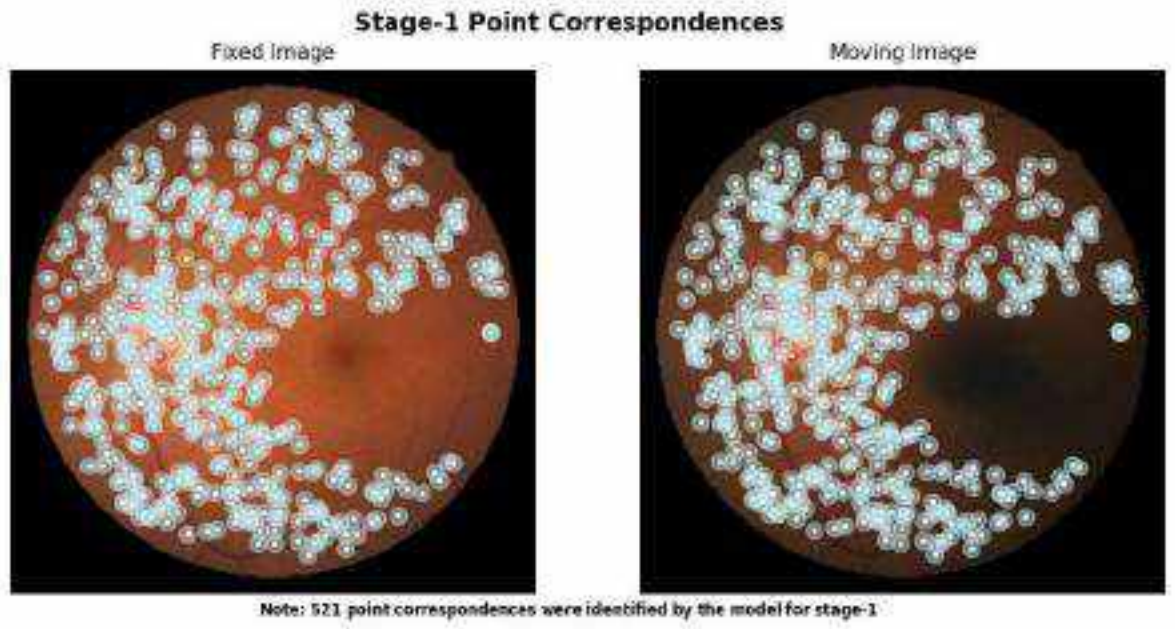

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



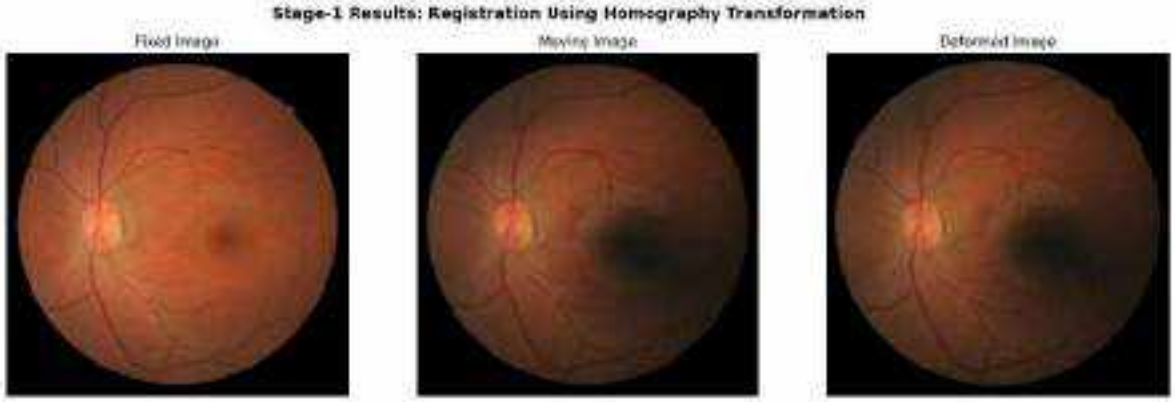
Note: 558 point correspondences were identified by the model for stage-2



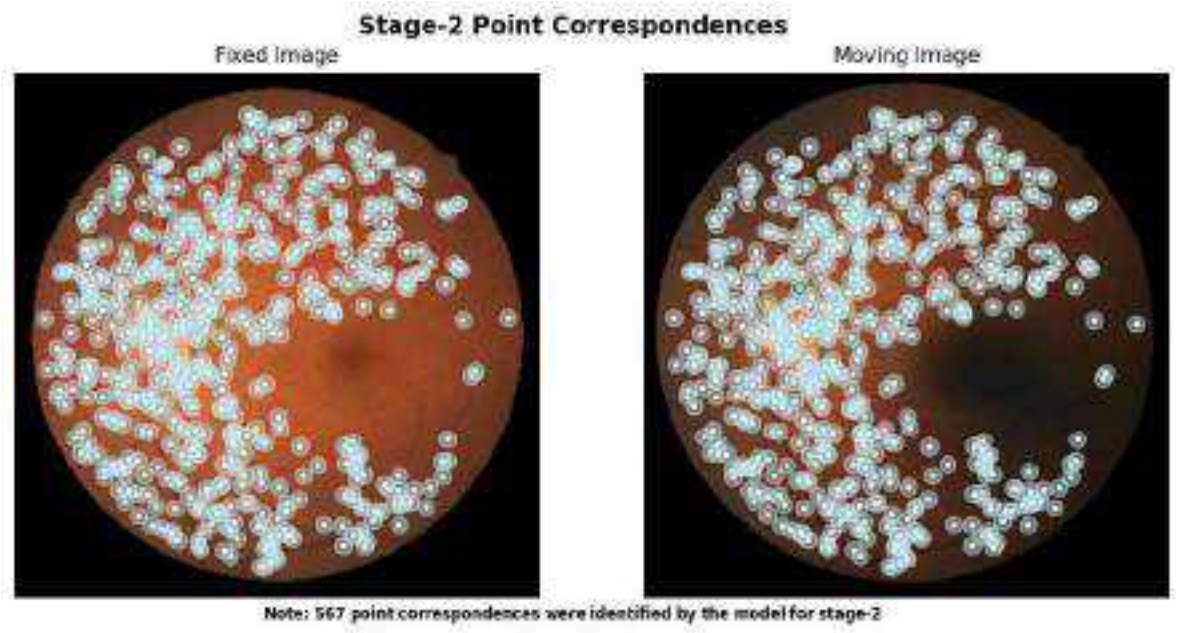
```
Mean Landmark Error for Case 9 Before Registration is 24.505014058030138 pixels
Mean Landmark Error for Case 9 After Registration is 1.598484524325156 pixels
Case 10
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S11_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S11_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

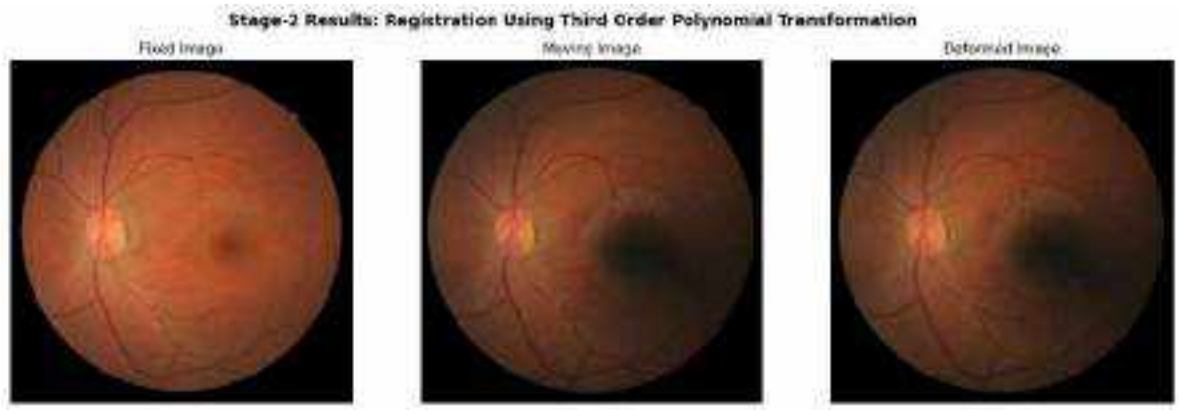


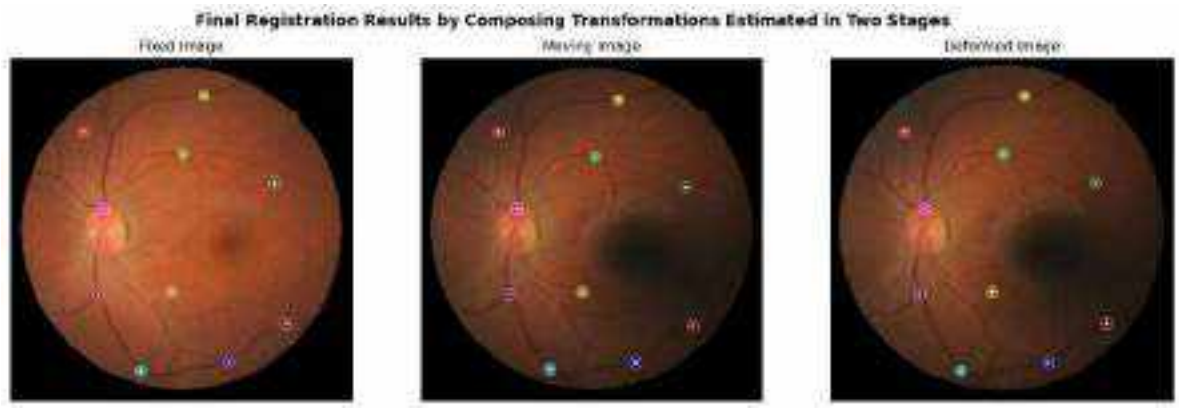

```
Homography Matrix:
[[ 1.01028310e+00  2.37696093e-02 -1.96936690e+01]
 [-2.45539745e-02  1.01392660e+00  5.00665106e-01]
 [-6.79544711e-06  1.84432431e-06  1.00000000e+00]]
```



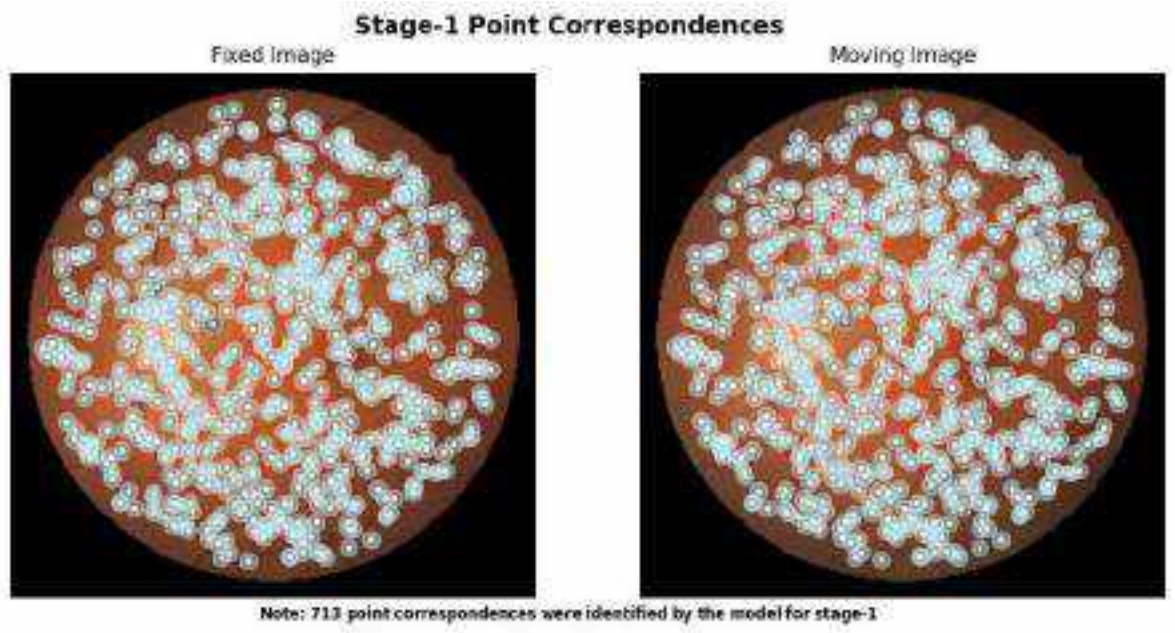

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





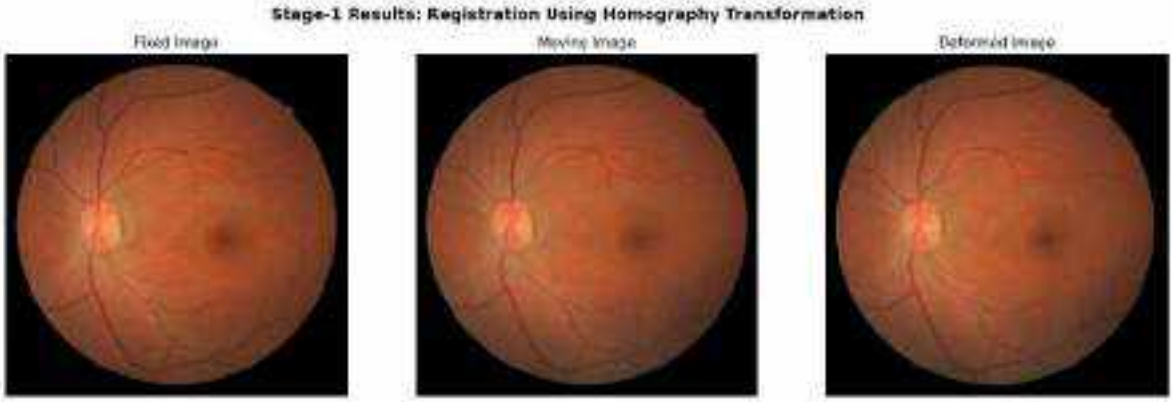



```
Mean Landmark Error for Case 10 Before Registration is 26.931720529141625 pixels
Mean Landmark Error for Case 10 After Registration is 1.7177497530550583 pixels
Case 11
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S12_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S12_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

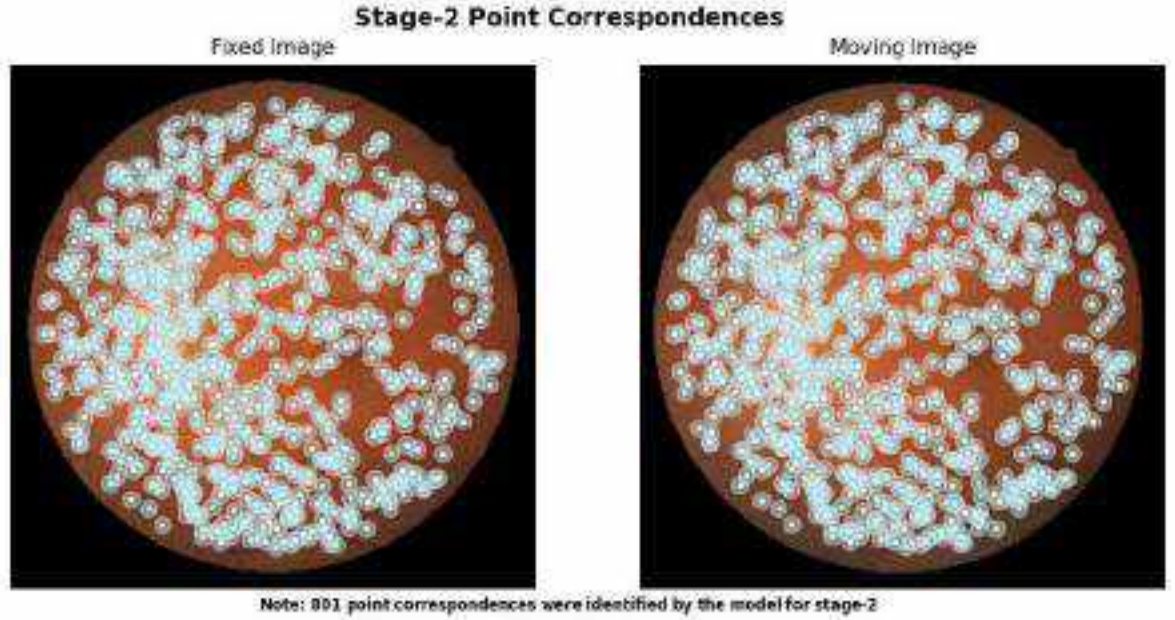


Note: 713 point correspondences were identified by the model for stage-1

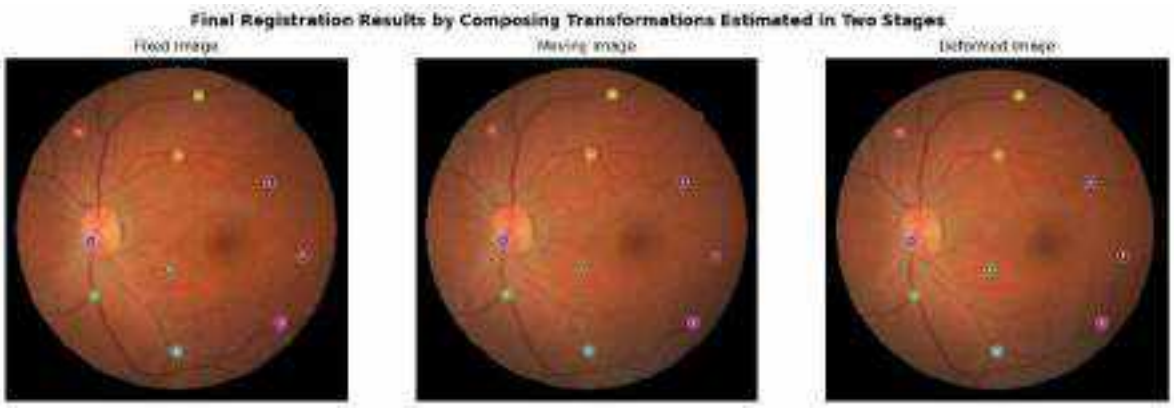
```
Homography Matrix:
[[ 9.94350879e-01  2.59948615e-03 -5.84294074e+00]
 [-3.06605054e-03  9.94454349e-01  5.24327082e+00]
 [-1.36474005e-06 -1.92182536e-06  1.00000000e+00]]
```



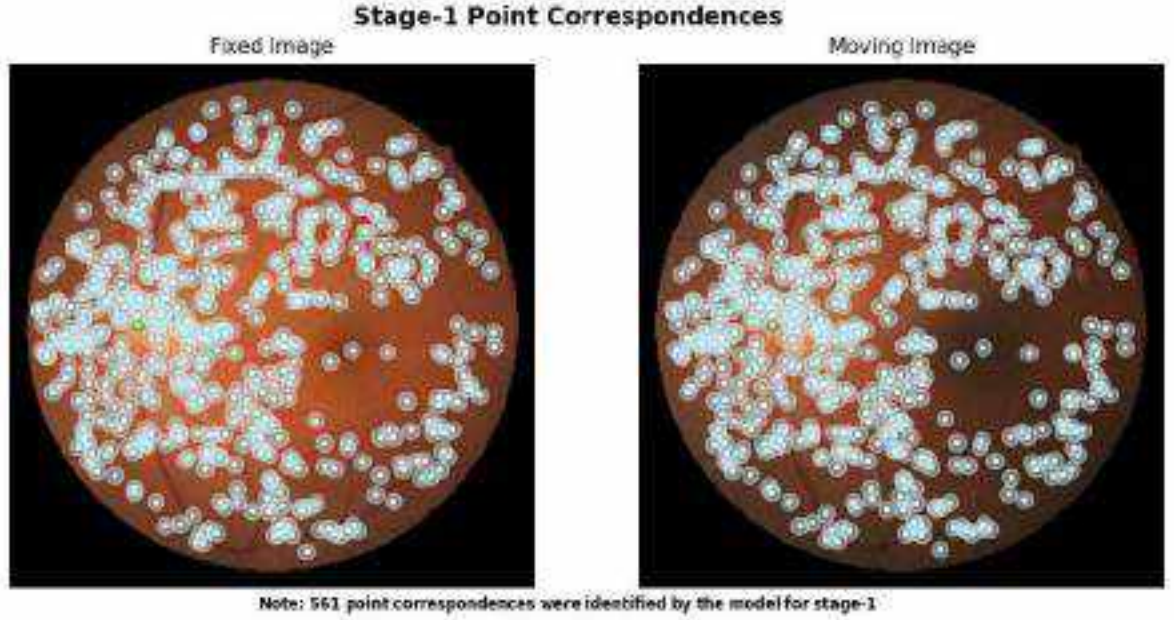

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



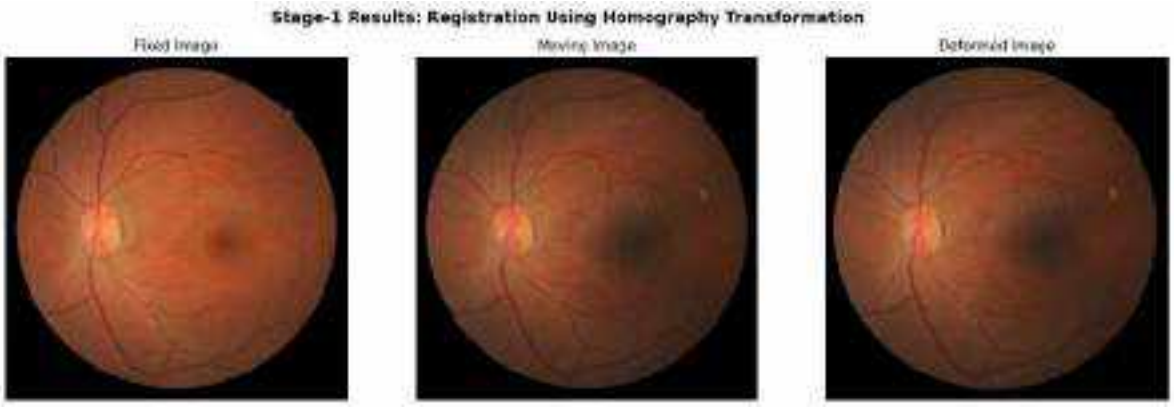
Note: 801 point correspondences were identified by the model for stage-2



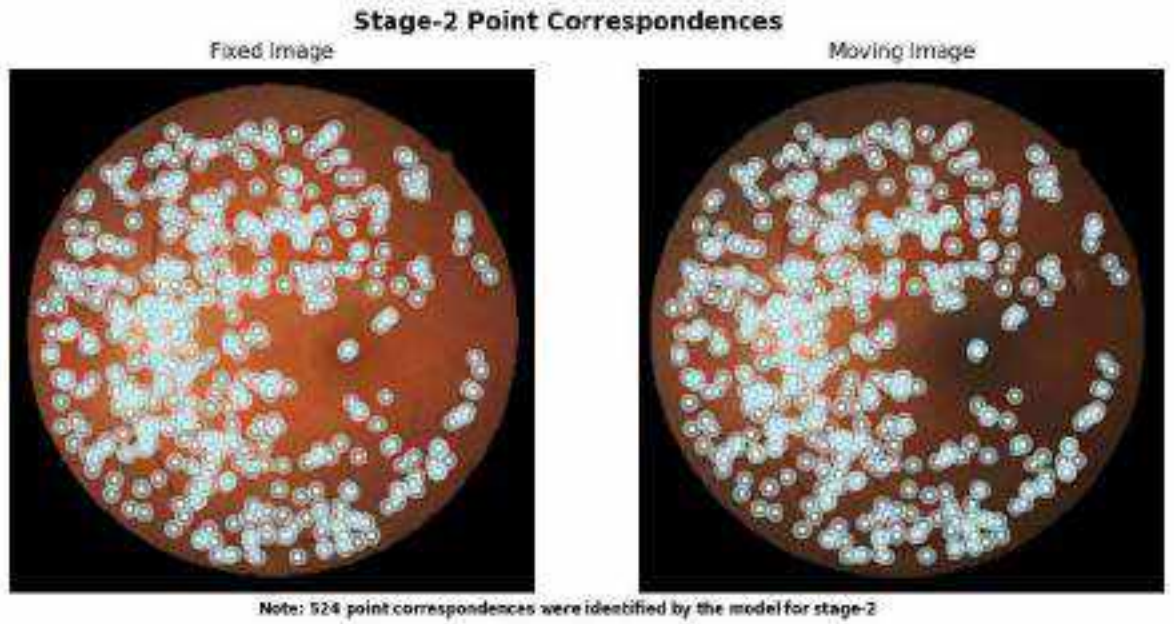

```
Mean Landmark Error for Case 11 Before Registration is 23.53937933071066 pixels
Mean Landmark Error for Case 11 After Registration is 1.19693960589496 pixels
Case 12
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S13_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S13_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



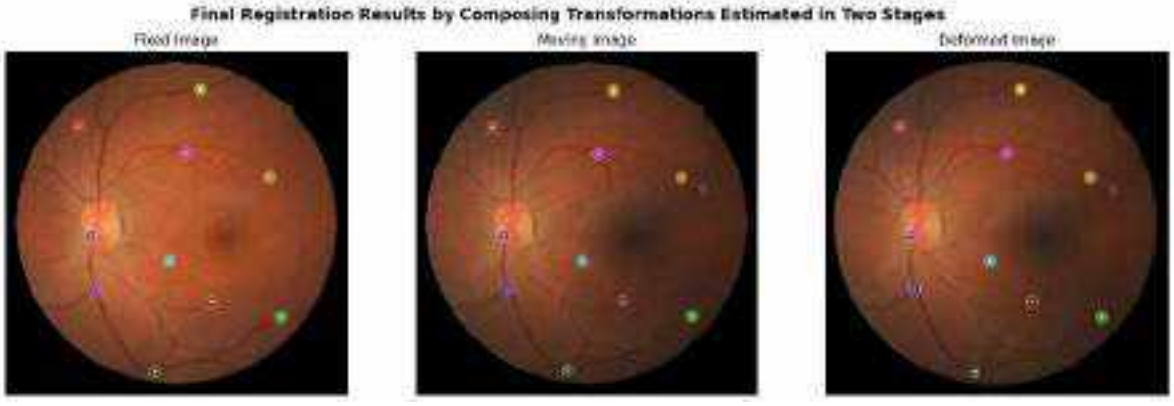

```
Homography Matrix:
[[ 1.00808652e+00  1.93824020e-04 -1.03077079e+01]
 [-3.92861256e-04  1.00755841e+00 -3.80550867e+00]
 [-1.72352405e-06  1.81391282e-07  1.00000000e+00]]
```



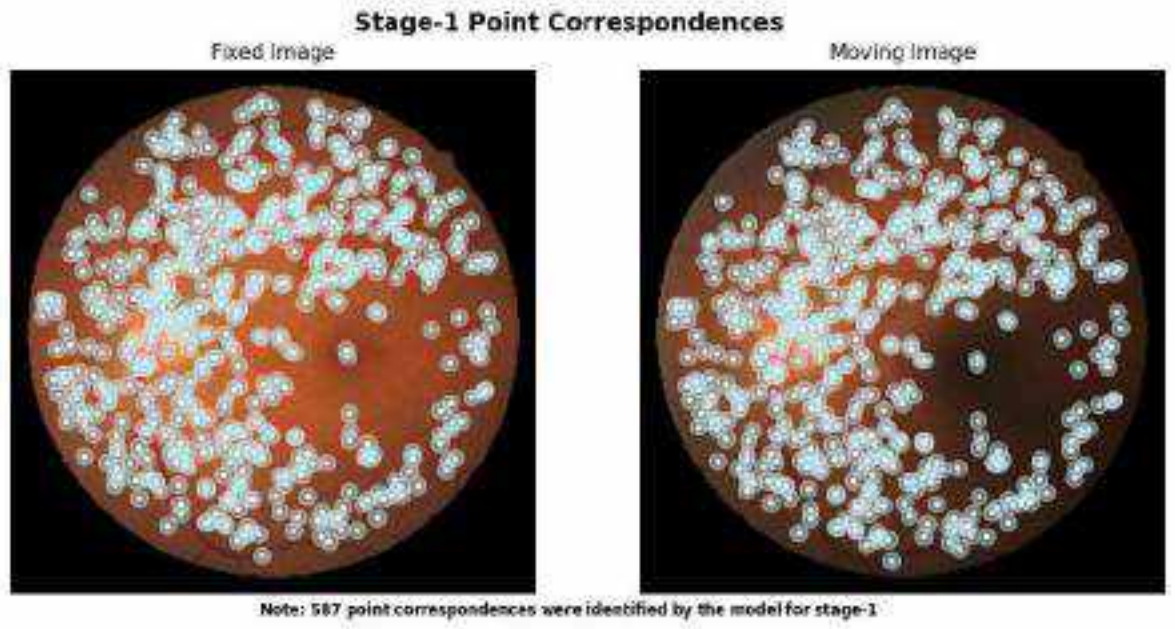

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



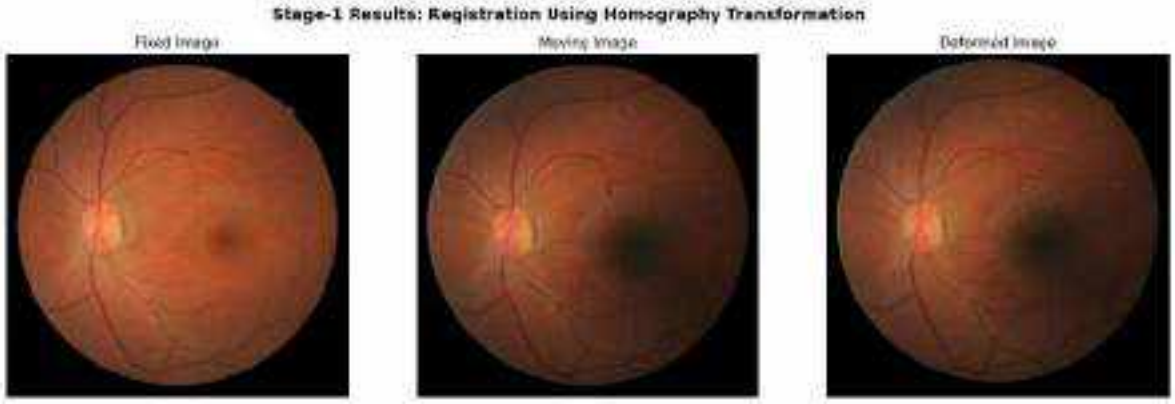
Note: 524 point correspondences were identified by the model for stage-2



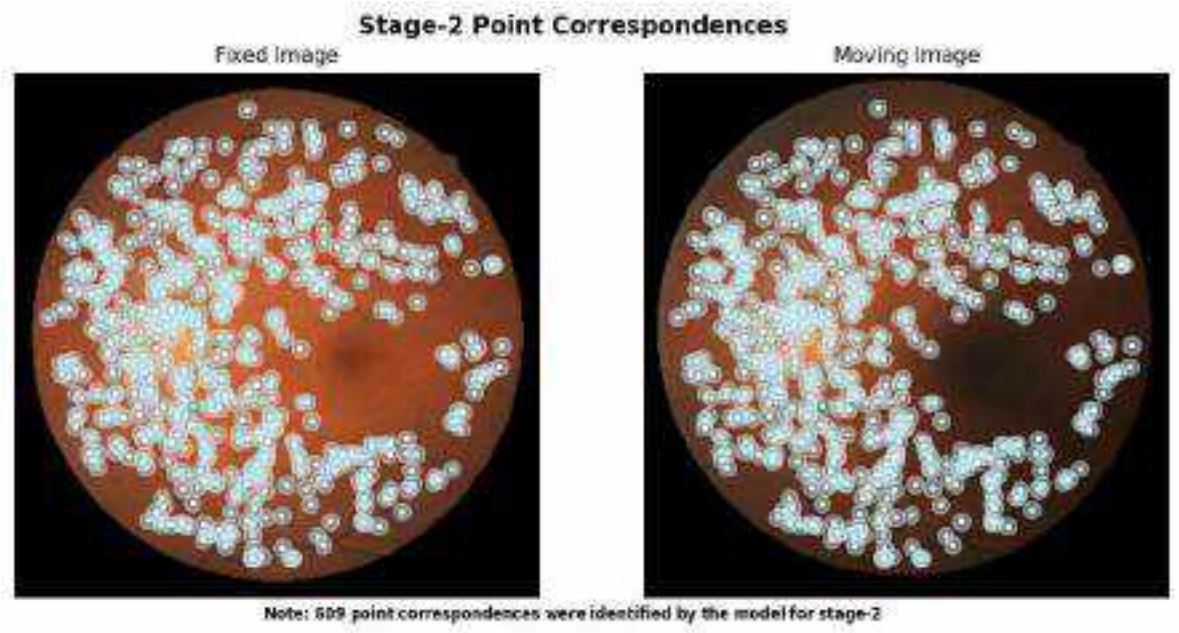
```
Mean Landmark Error for Case 12 Before Registration is 21.816476137741642 pixels
Mean Landmark Error for Case 12 After Registration is 1.7696157959260717 pixels
Case 13
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S14_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S14_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



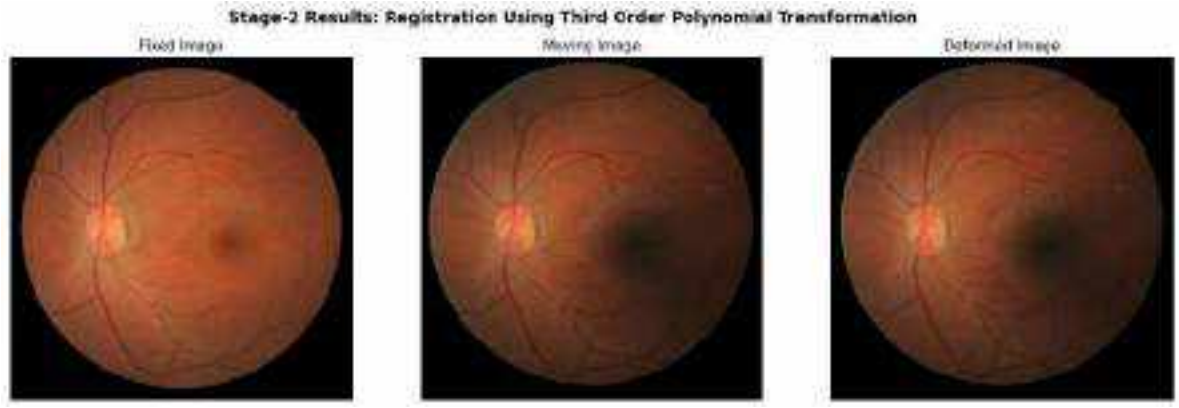

```
Homography Matrix:
[[ 1.00797694e+00 -3.02683278e-04 -5.66744878e+00]
 [-8.68066377e-04  1.00457066e+00 -1.62124780e+01]
 [ 4.68867111e-06 -7.24373206e-06  1.00000000e+00]]
```

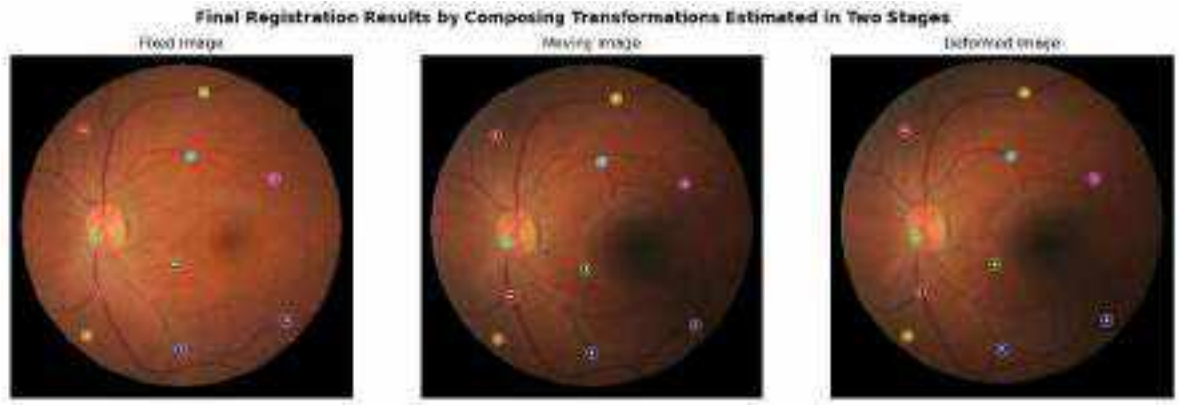



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

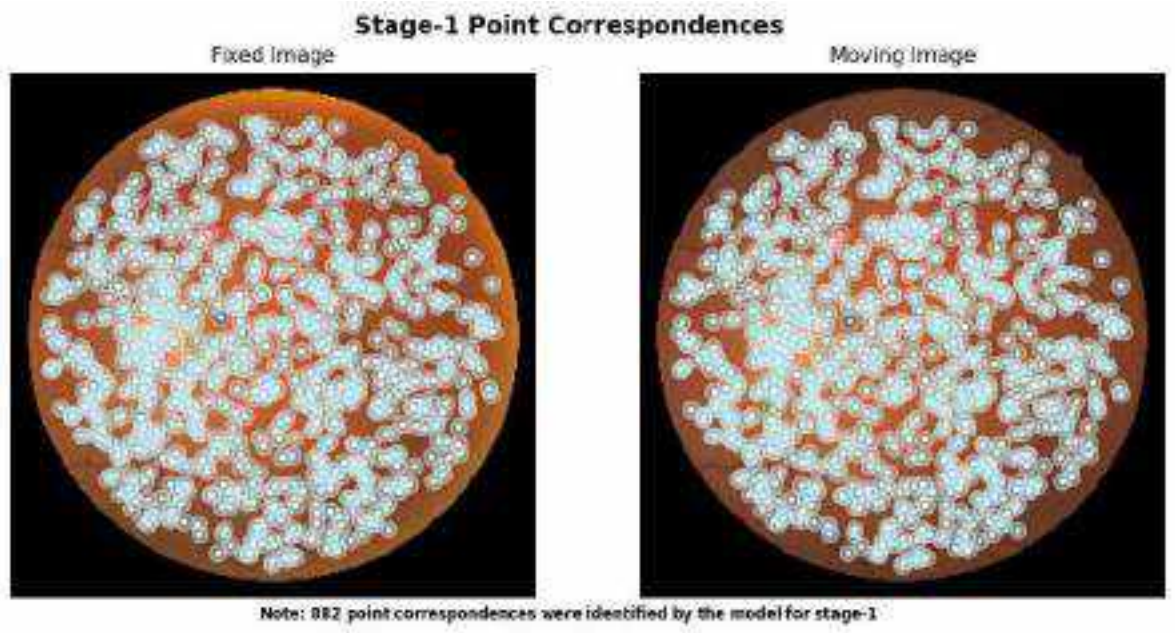


Note: 509 point correspondences were identified by the model for stage-2





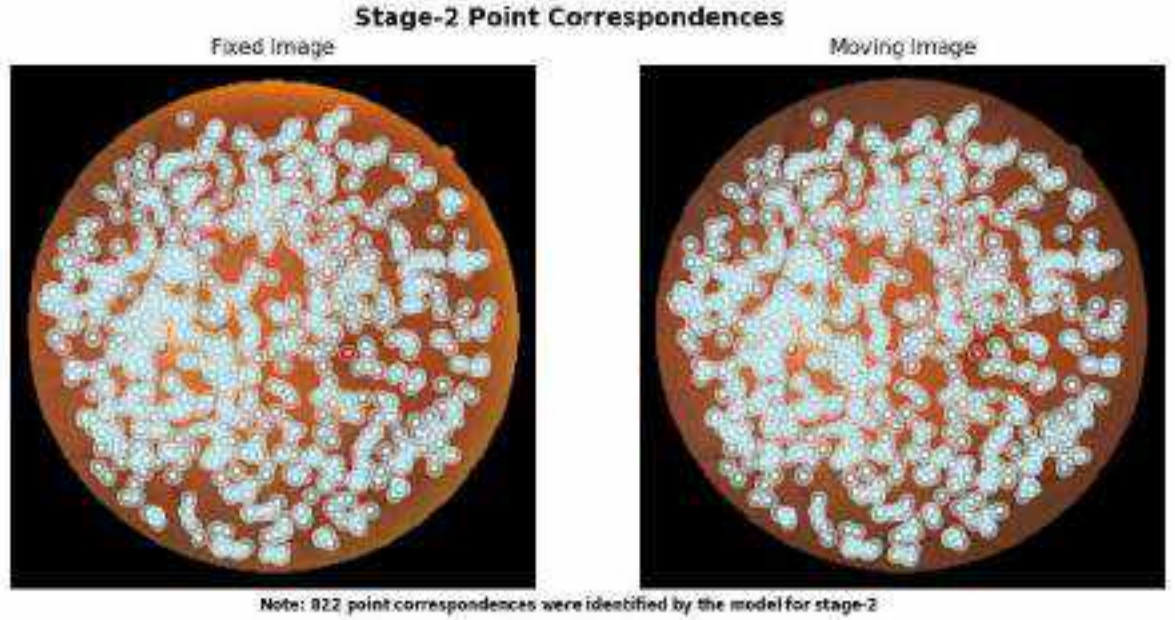

```
Mean Landmark Error for Case 13 Before Registration is 42.09029827008554 pixels
Mean Landmark Error for Case 13 After Registration is 1.4893707547093962 pixels
Case 14
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S15_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S15_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



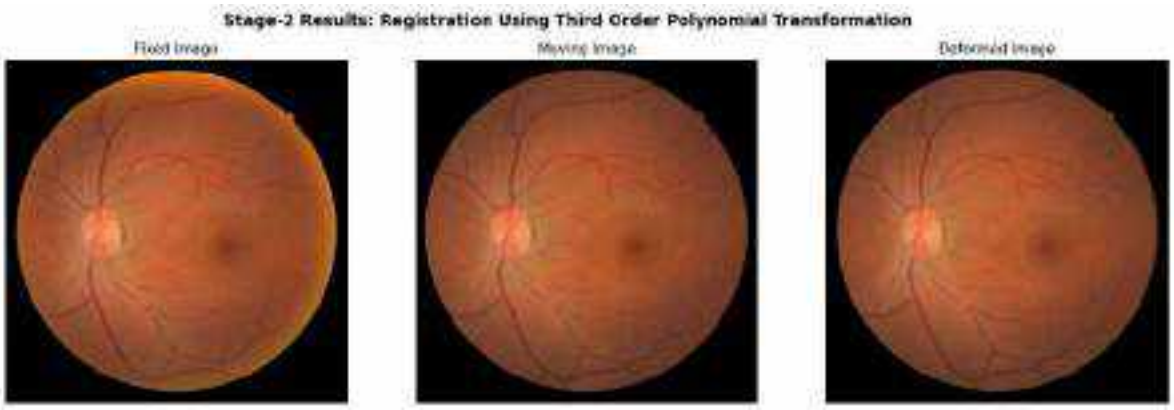
Note: 882 point correspondences were identified by the model for stage-1

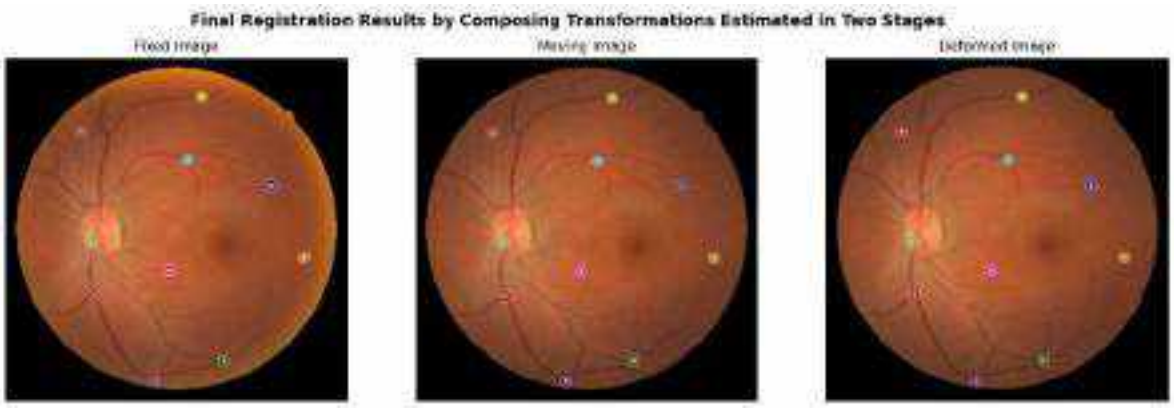
```
Homography Matrix:
[[ 1.00390092e+00 -9.61534779e-04 -2.34038797e+00]
 [ 6.90887279e-04  1.00321152e+00 -4.15589620e+00]
 [ 8.90655297e-08 -1.44216521e-06  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

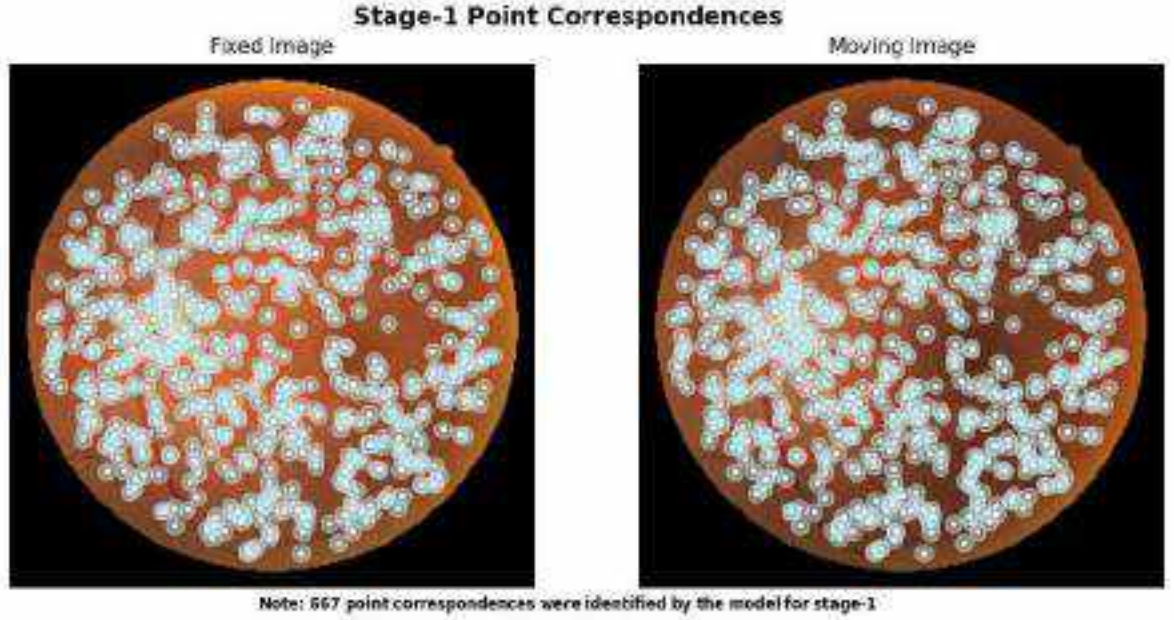


Note: 822 point correspondences were identified by the model for stage-2

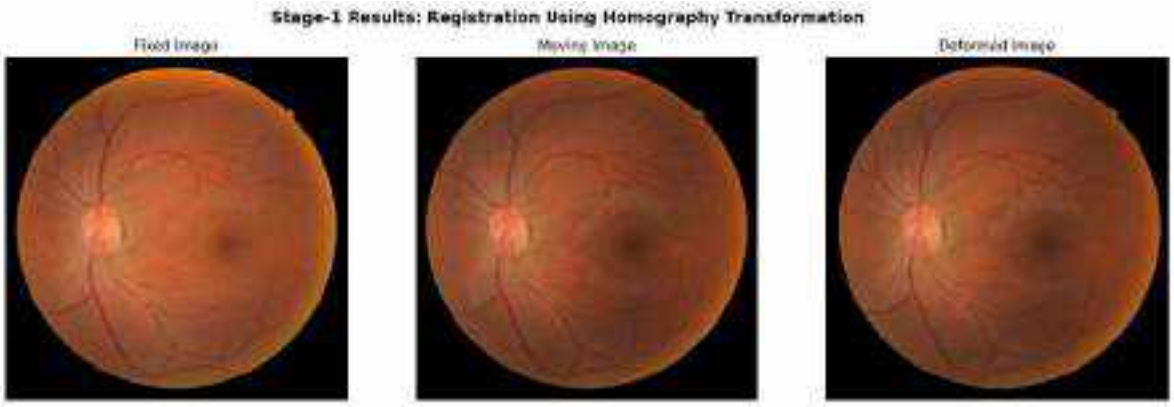




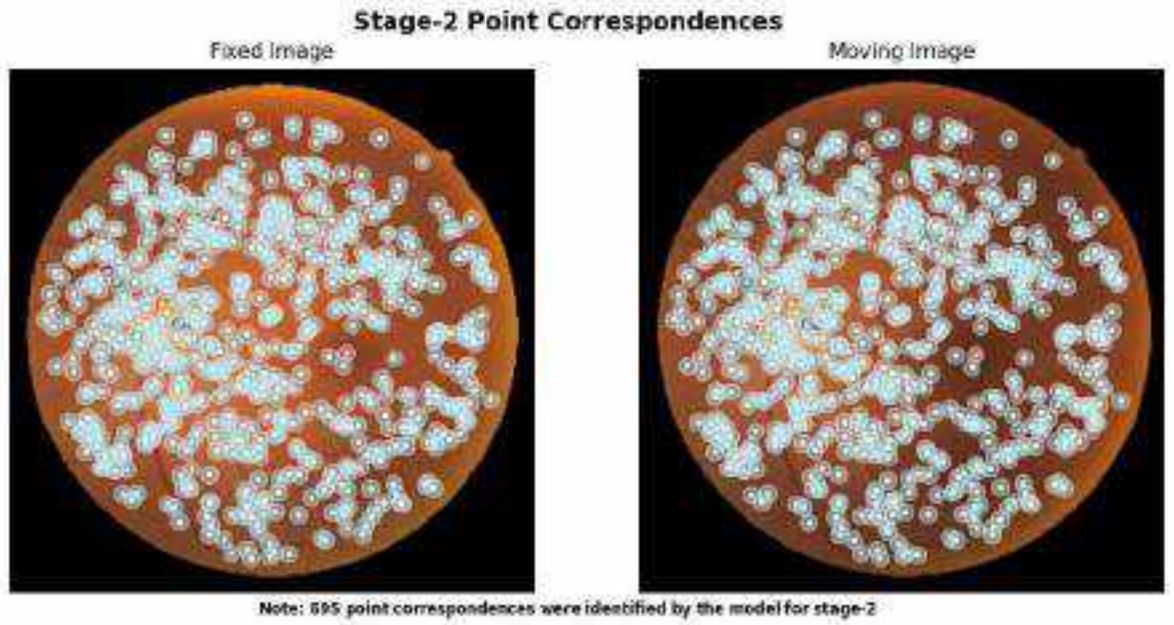

```
Mean Landmark Error for Case 14 Before Registration is 9.007109412574113 pixels
Mean Landmark Error for Case 14 After Registration is 0.9954089741395571 pixels
Case 15
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S16_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S16_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



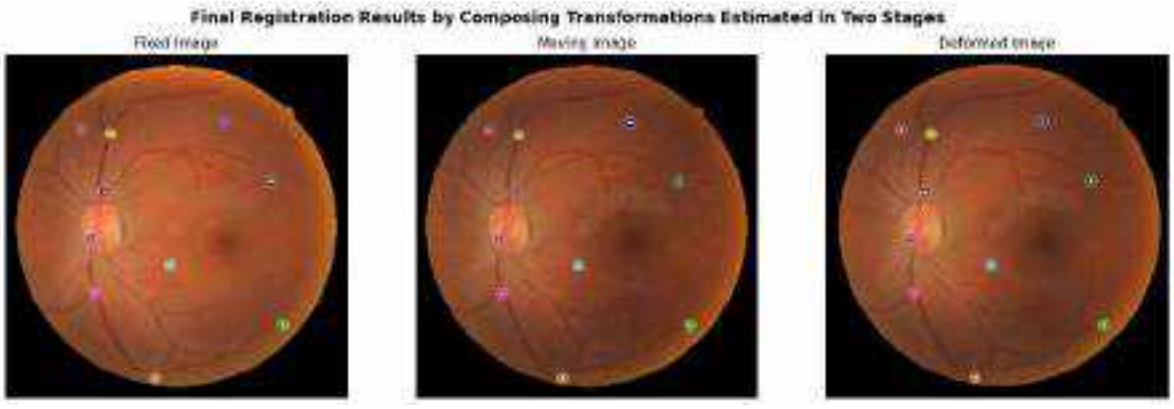

```
Homography Matrix:
[[ 1.00808484e+00  1.41067075e-03  2.40137203e+00]
 [ 1.41040195e-04  1.00562925e+00 -4.58933496e+00]
 [ 2.36161103e-06 -5.08852212e-07  1.00000000e+00]]
```



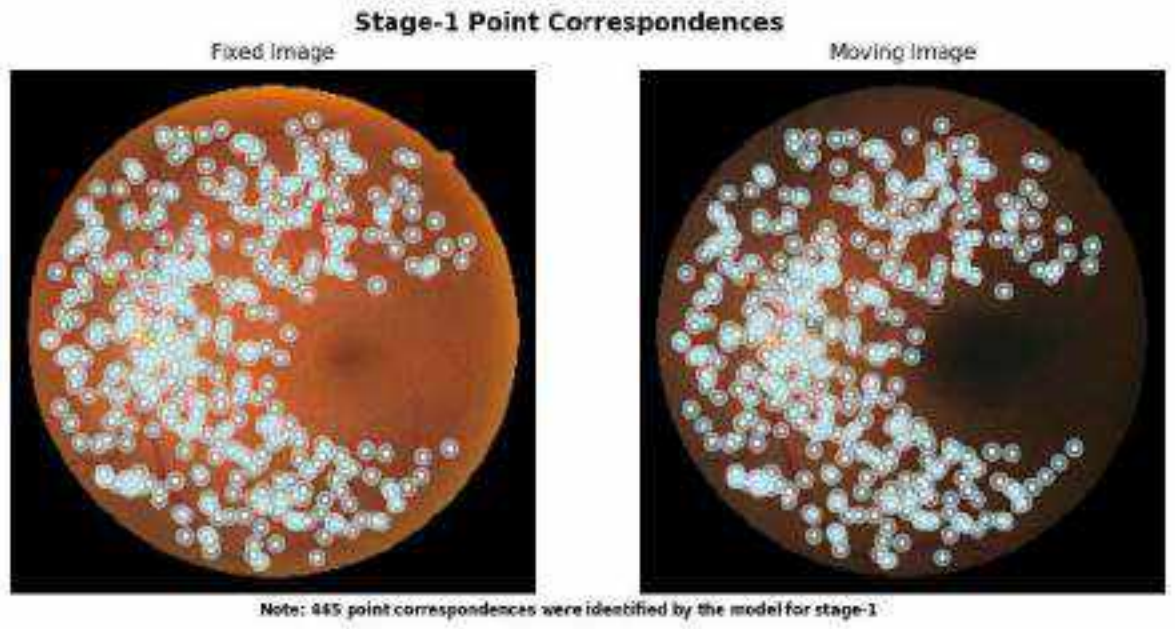

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



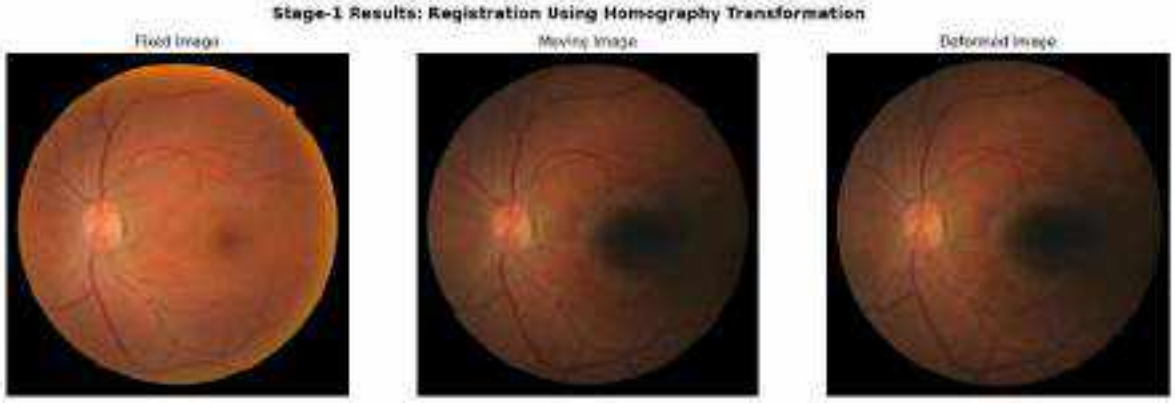
Note: 595 point correspondences were identified by the model for stage-2



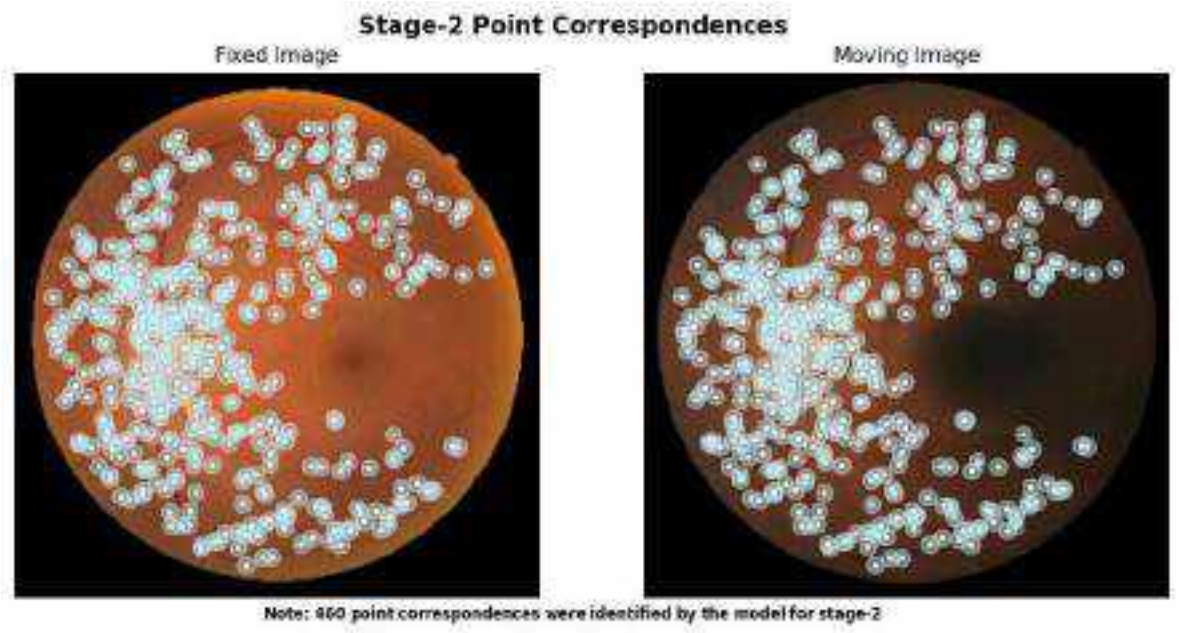
```
Mean Landmark Error for Case 15 Before Registration is 22.391698212583946 pixels
Mean Landmark Error for Case 15 After Registration is 1.3799992321968007 pixels
Case 16
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S17_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S17_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

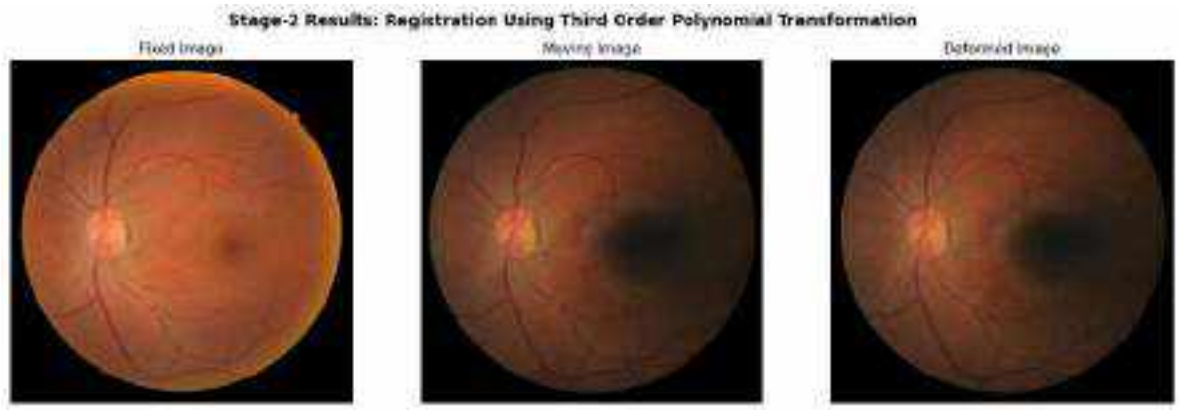


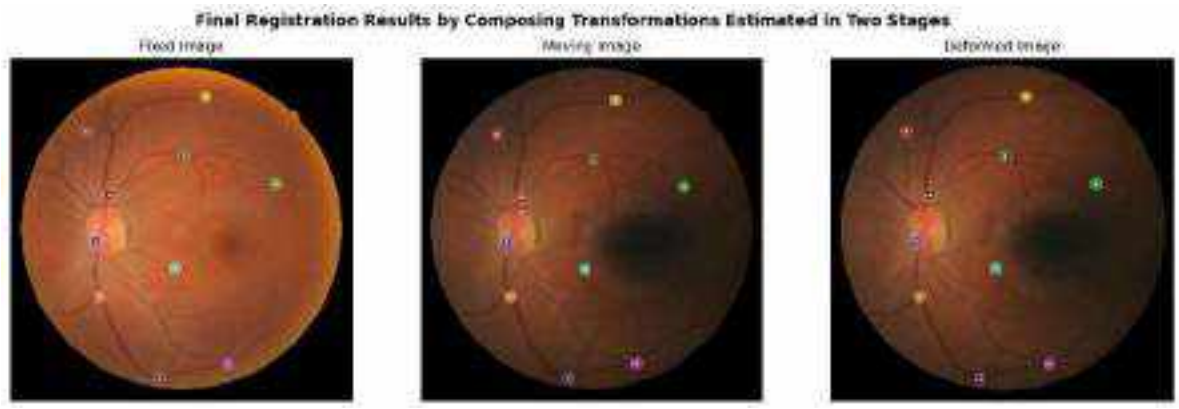

```
Homography Matrix:
[[ 1.01499696e+00  4.57126694e-04 -5.63568952e+00]
 [-3.62764318e-03  1.01437549e+00 -1.05464004e+01]
 [ 9.62127552e-08 -4.09084962e-06  1.00000000e+00]]
```



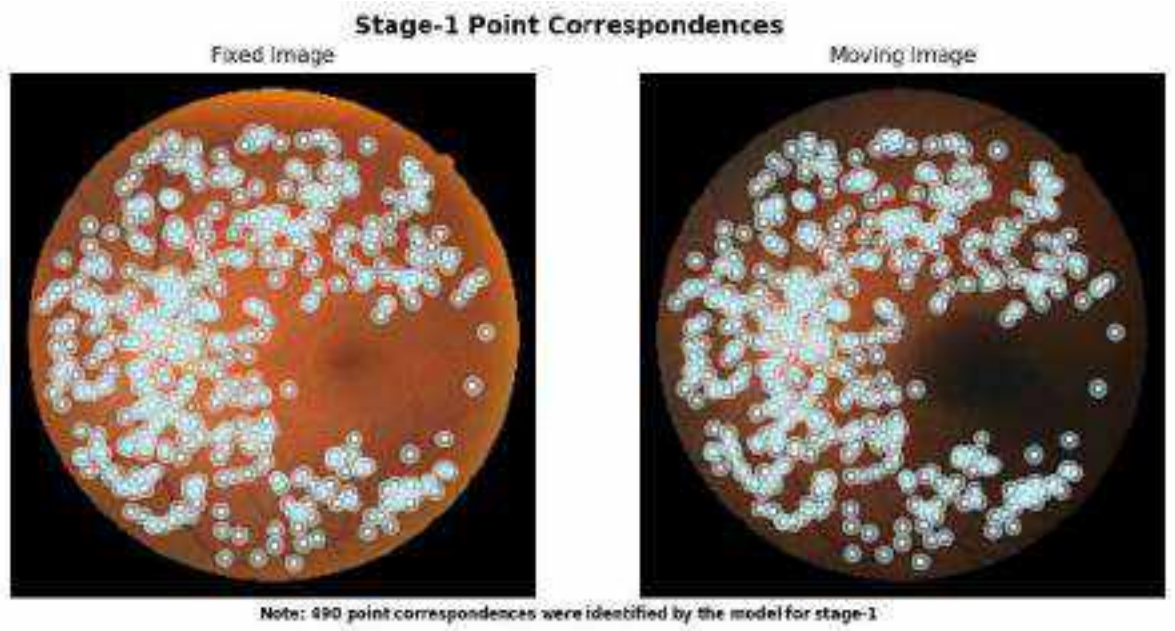

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





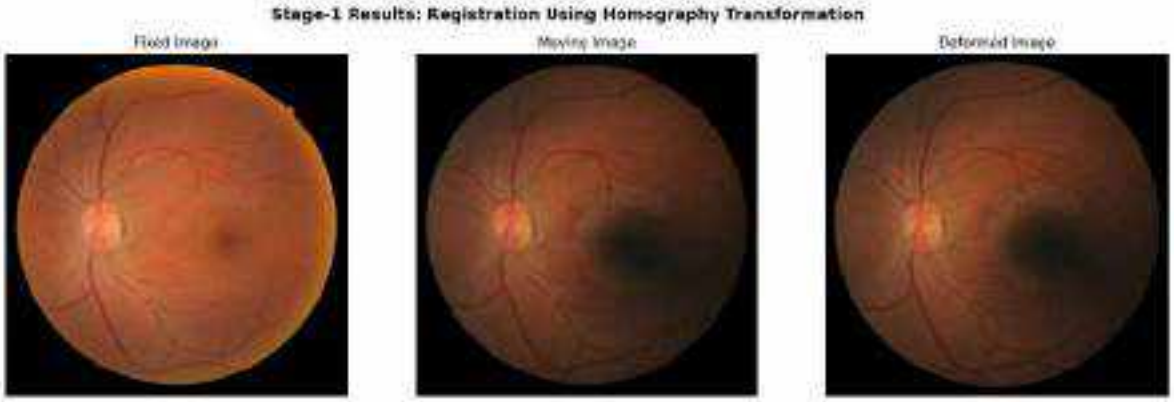



```
Mean Landmark Error for Case 16 Before Registration is 19.268123314360167 pixels
Mean Landmark Error for Case 16 After Registration is 2.1236769979090043 pixels
Case 17
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S18_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S18_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

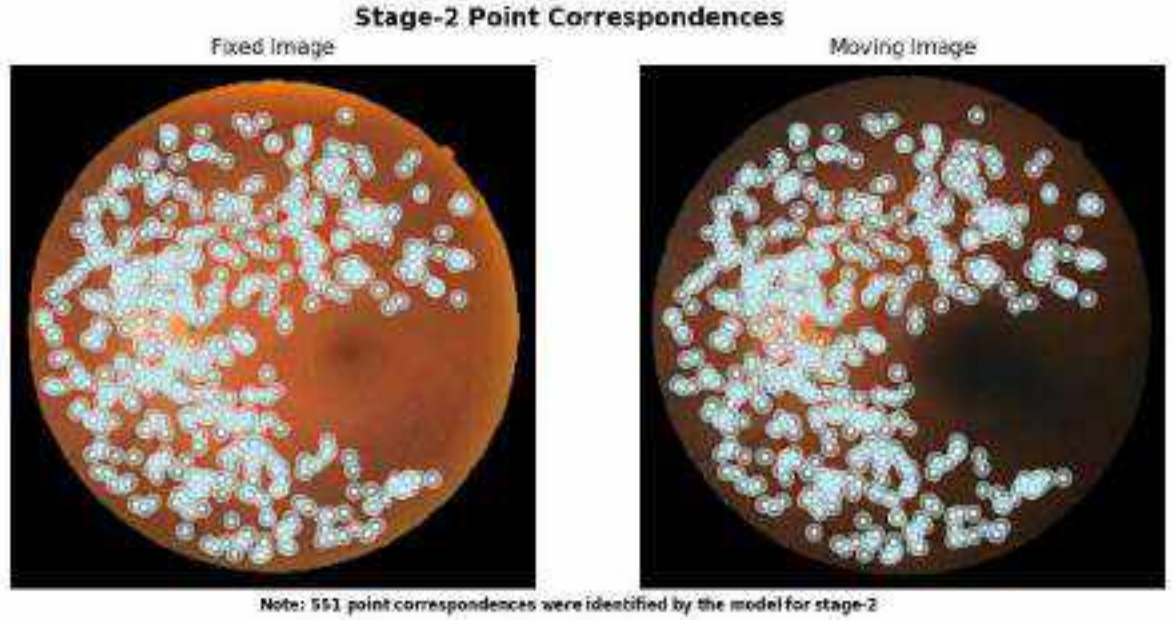


Note: 690 point correspondences were identified by the model for stage-1

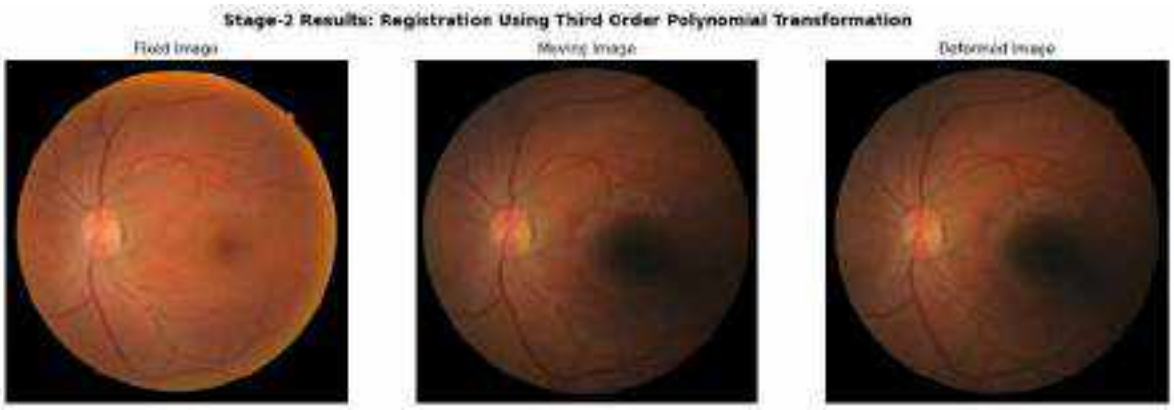
```
Homography Matrix:
[[ 1.02153568e+00  1.12530790e-02 -1.37452665e+01]
 [-8.61954392e-03  1.01889870e+00 -5.20727908e+00]
 [ 5.29057056e-06 -1.07036430e-06  1.00000000e+00]]
```

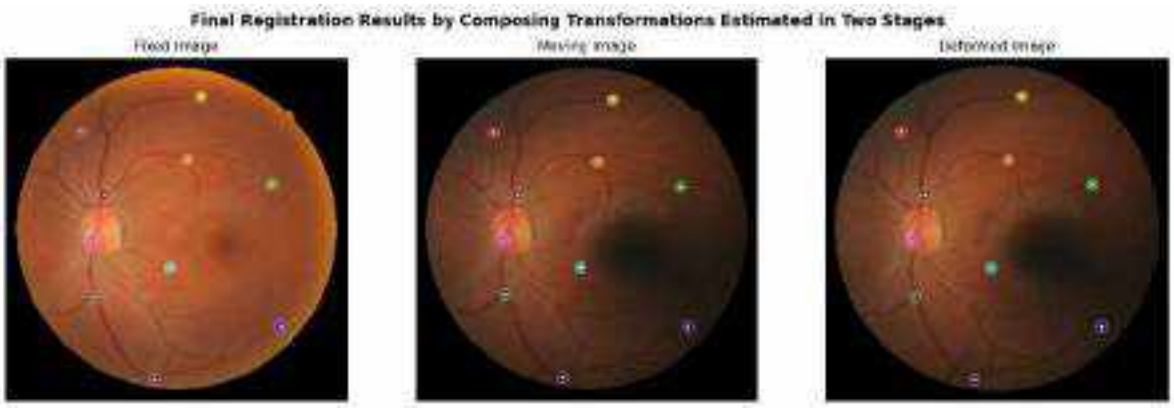



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

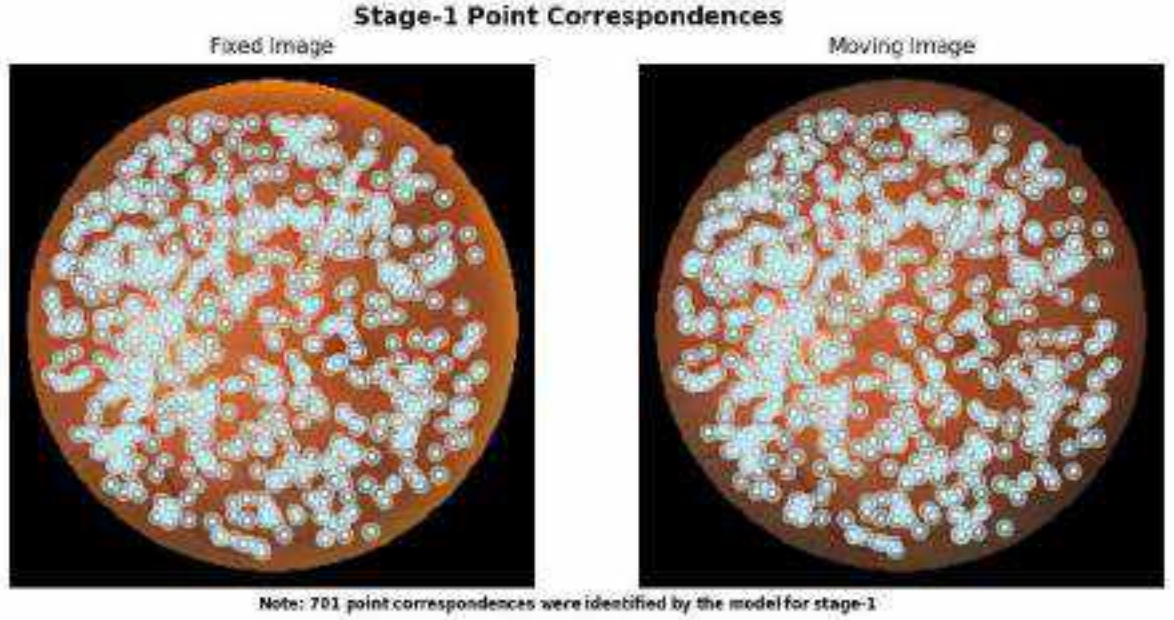


Note: 551 point correspondences were identified by the model for stage-2

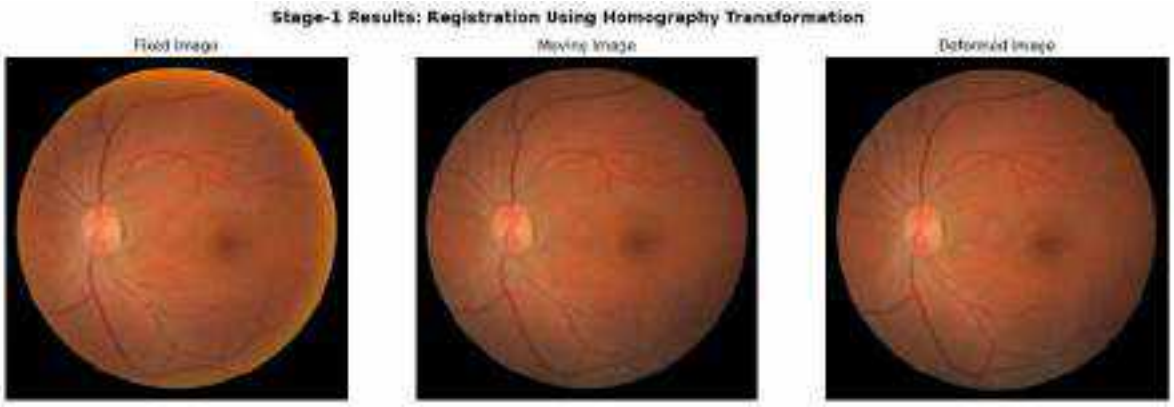




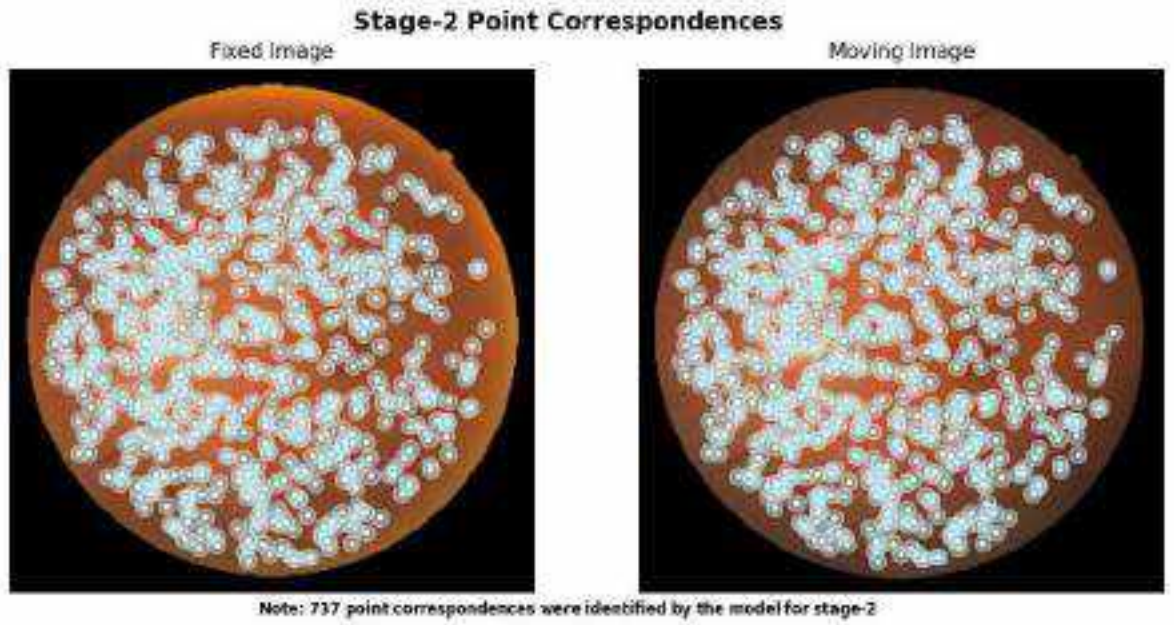

```
Mean Landmark Error for Case 17 Before Registration is 19.26269286694786 pixels
Mean Landmark Error for Case 17 After Registration is 1.9291882321976828 pixels
Case 18
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S19_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S19_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



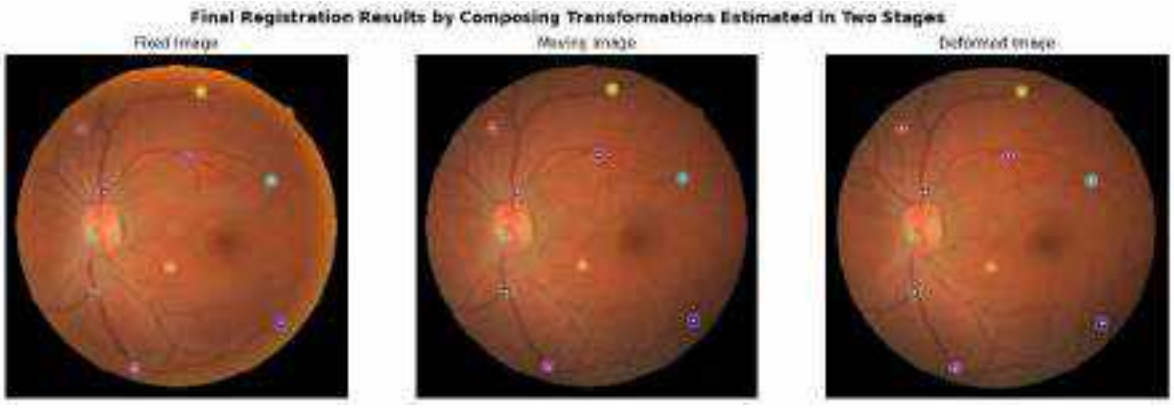

```
Homography Matrix:
[[ 9.96809025e-01 -7.30415948e-03  1.08036374e+00]
 [ 8.24694297e-03  9.99407346e-01  7.30153341e-01]
 [-2.16684893e-06  1.11827015e-06  1.00000000e+00]]
```



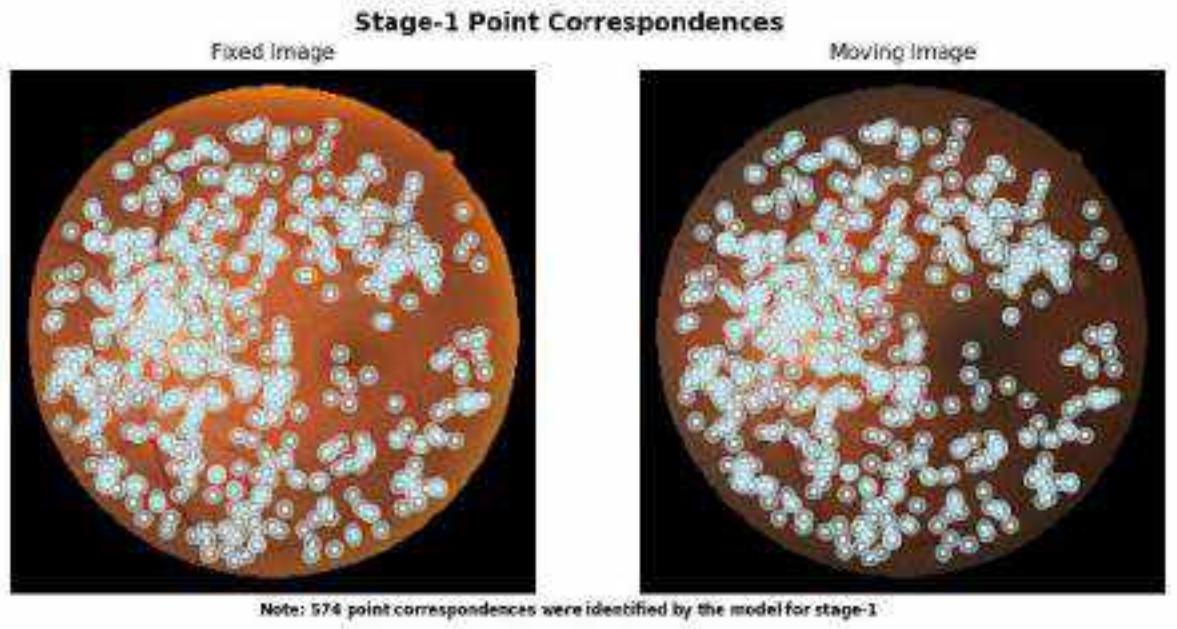

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



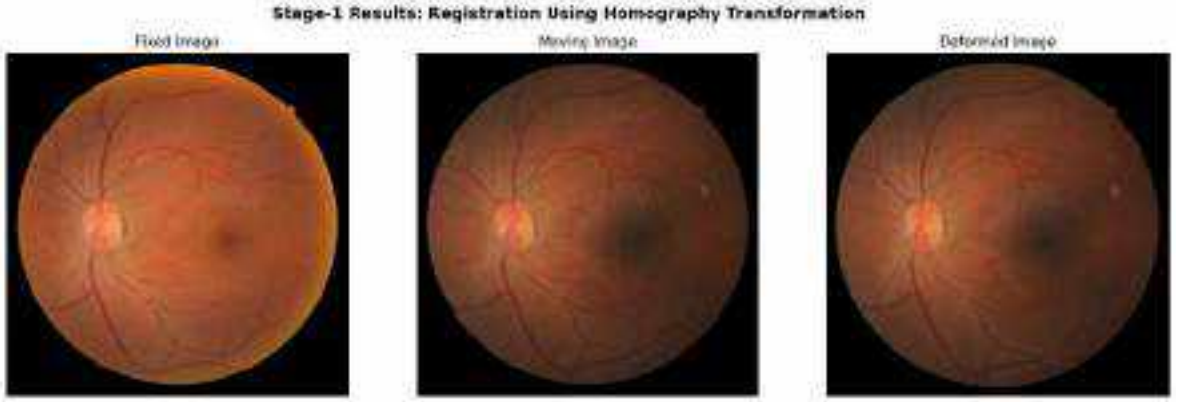
Note: 737 point correspondences were identified by the model for stage-2



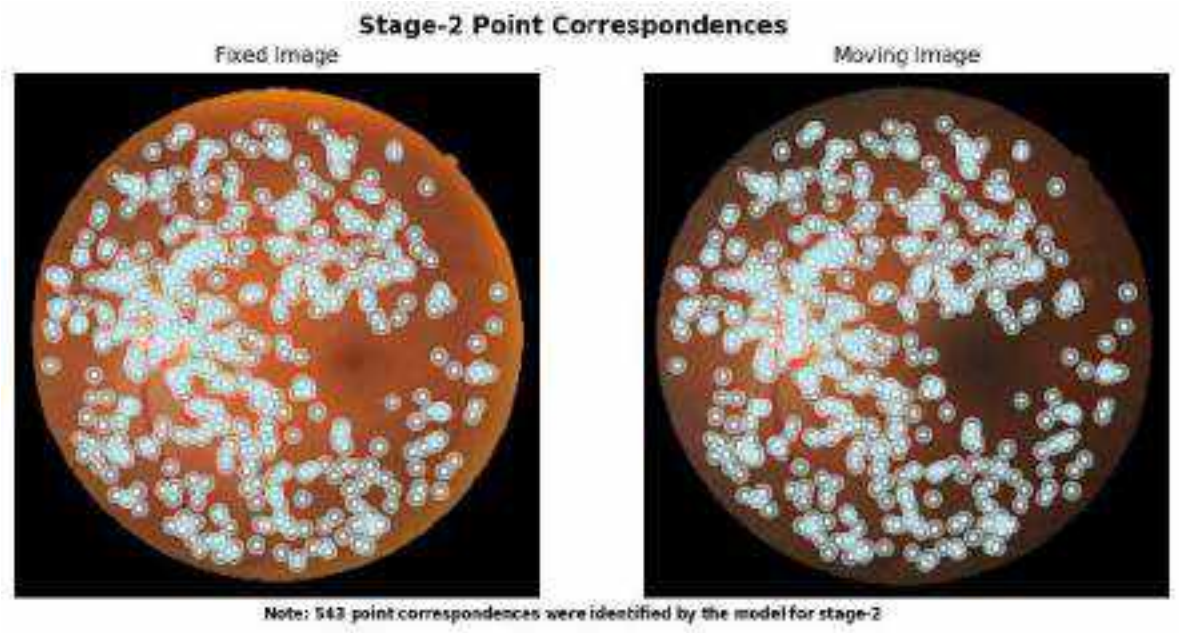
```
Mean Landmark Error for Case 18 Before Registration is 19.565399760185336 pixels
Mean Landmark Error for Case 18 After Registration is 1.3546114150256 35 pixels
Case 19
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S20_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S20_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

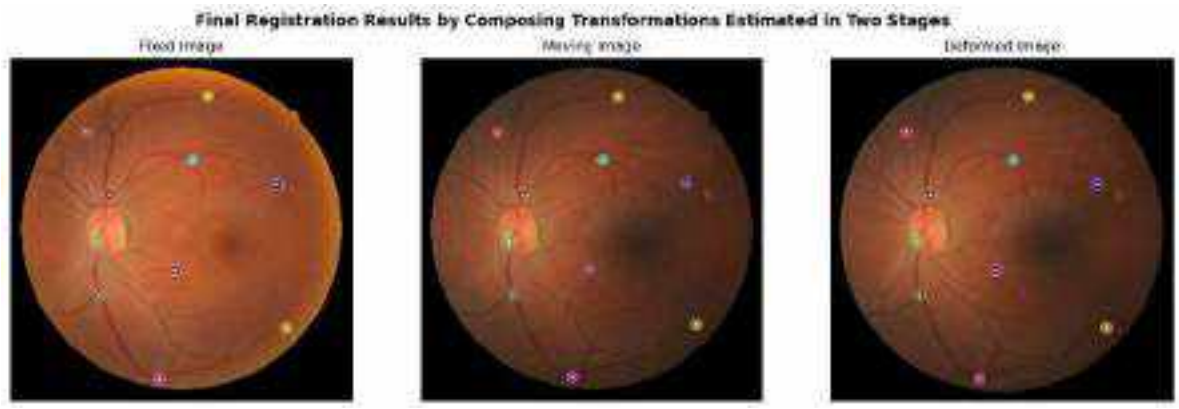



```
Homography Matrix:
[[ 1.01123479e+00 -1.10734935e-02 -3.35612767e+00]
 [ 1.18991962e-02  1.01037328e+00 -8.87376095e+00]
 [ 1.66927847e-06 -2.93144232e-06  1.00000000e+00]]
```

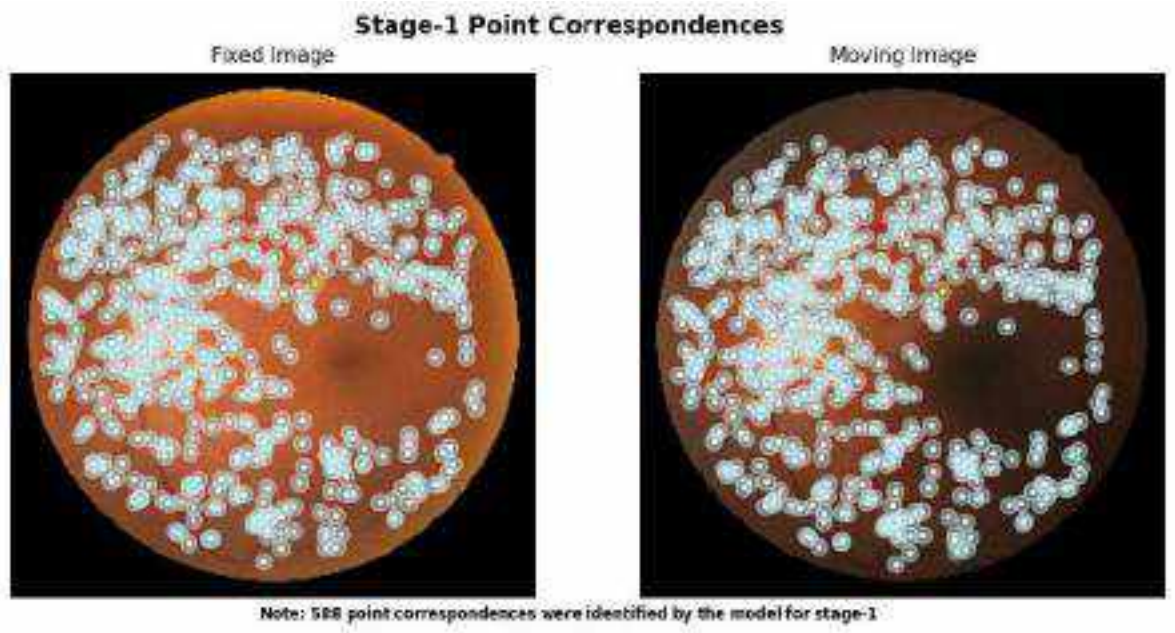



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



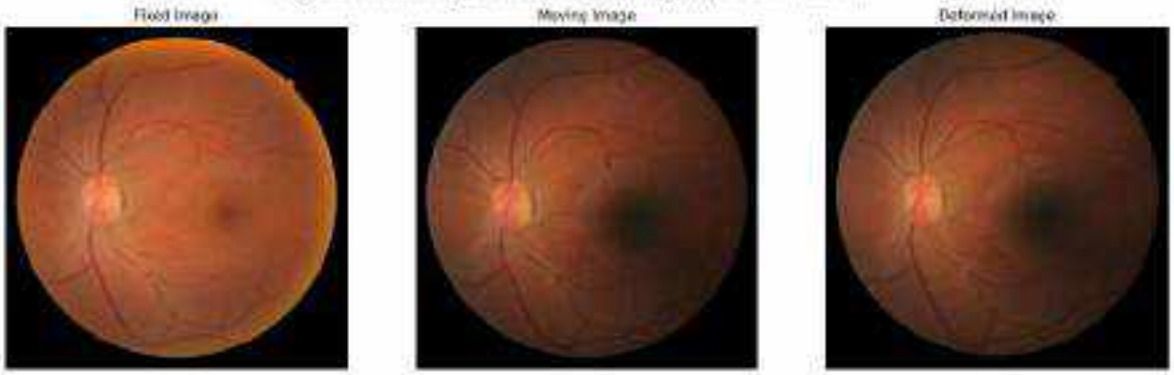



```
Mean Landmark Error for Case 19 Before Registration is 17.580299463191267 pixels
Mean Landmark Error for Case 19 After Registration is 2.114600337297947 pixels
Case 20
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S21_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S21_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

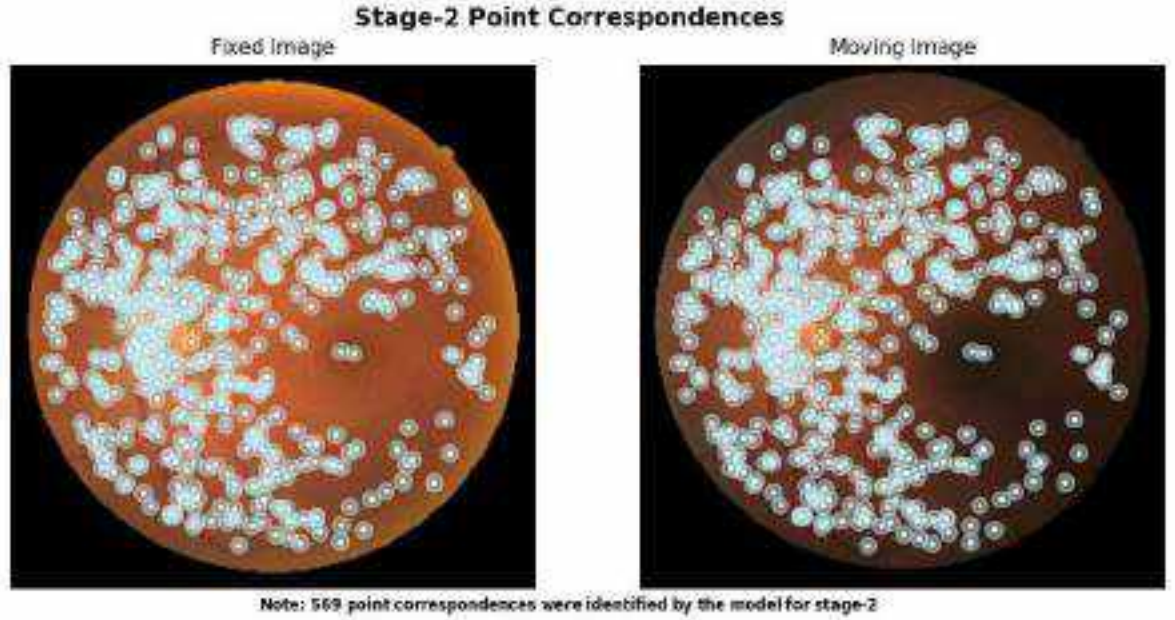


Note: 588 point correspondences were identified by the model for stage-1

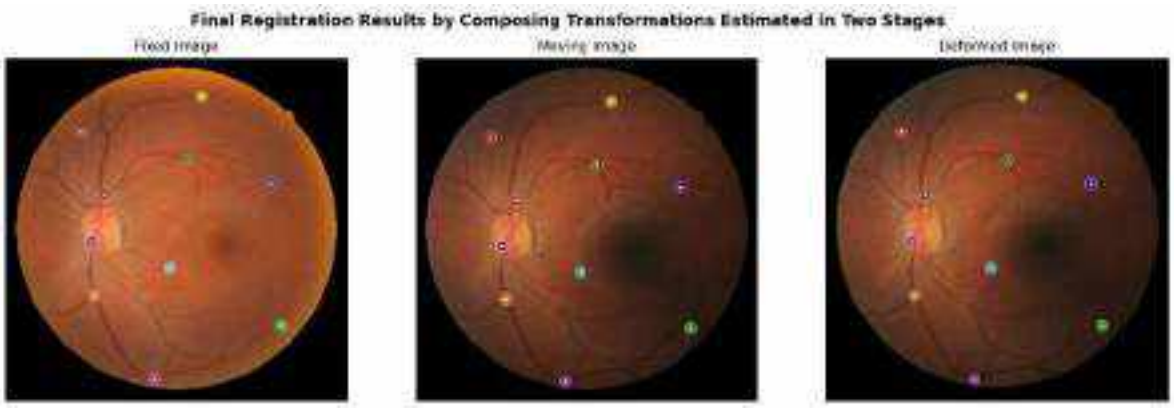
```
Homography Matrix:
[[ 1.00425169e+00 -1.14218864e-02  1.67659115e+00]
 [ 7.59550854e-03  1.00067366e+00 -1.81503932e+01]
 [ 1.20398672e-06 -1.26712828e-05  1.00000000e+00]]
```



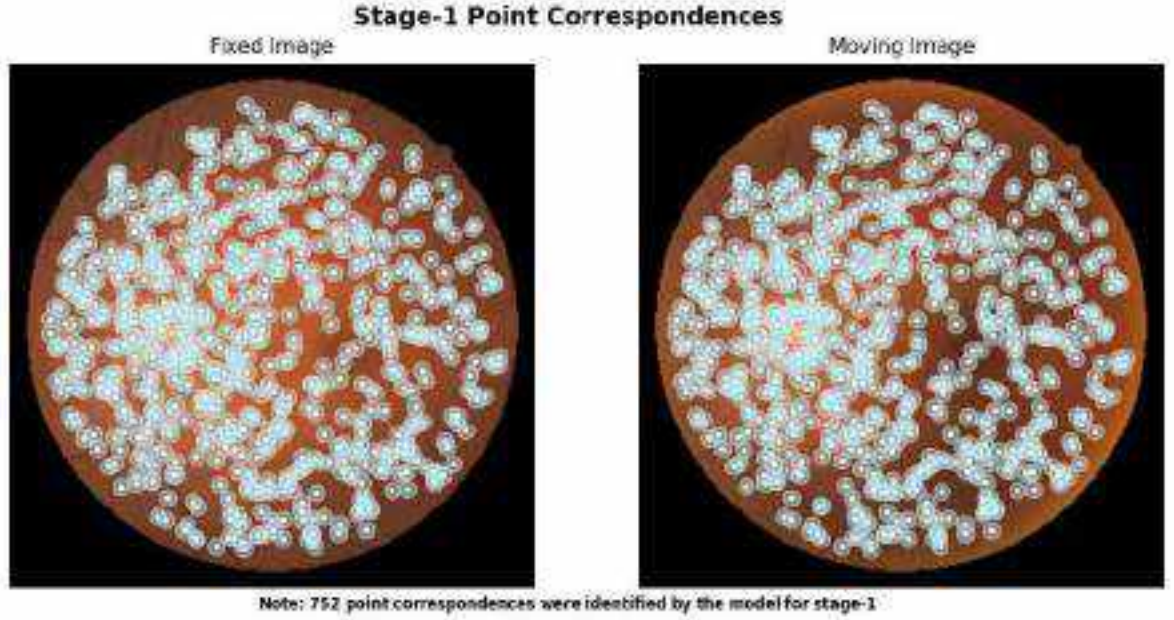

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



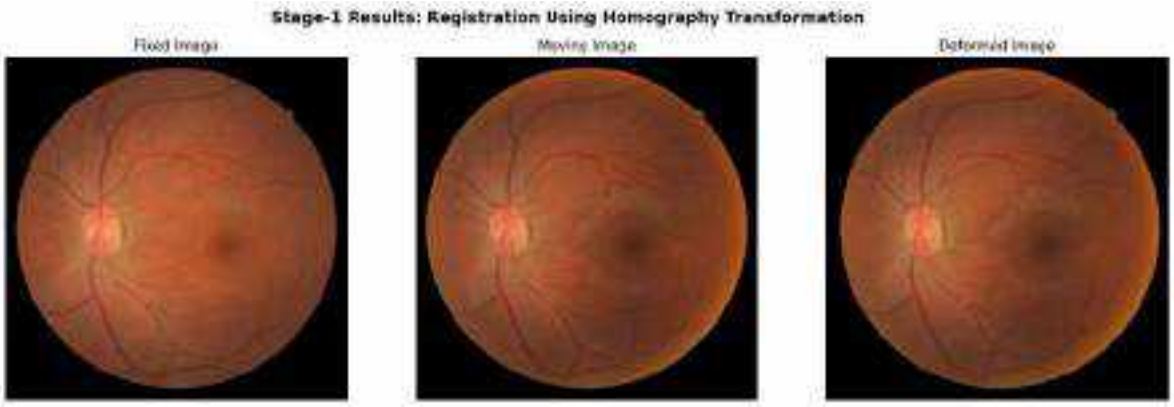
Note: 569 point correspondences were identified by the model for stage-2



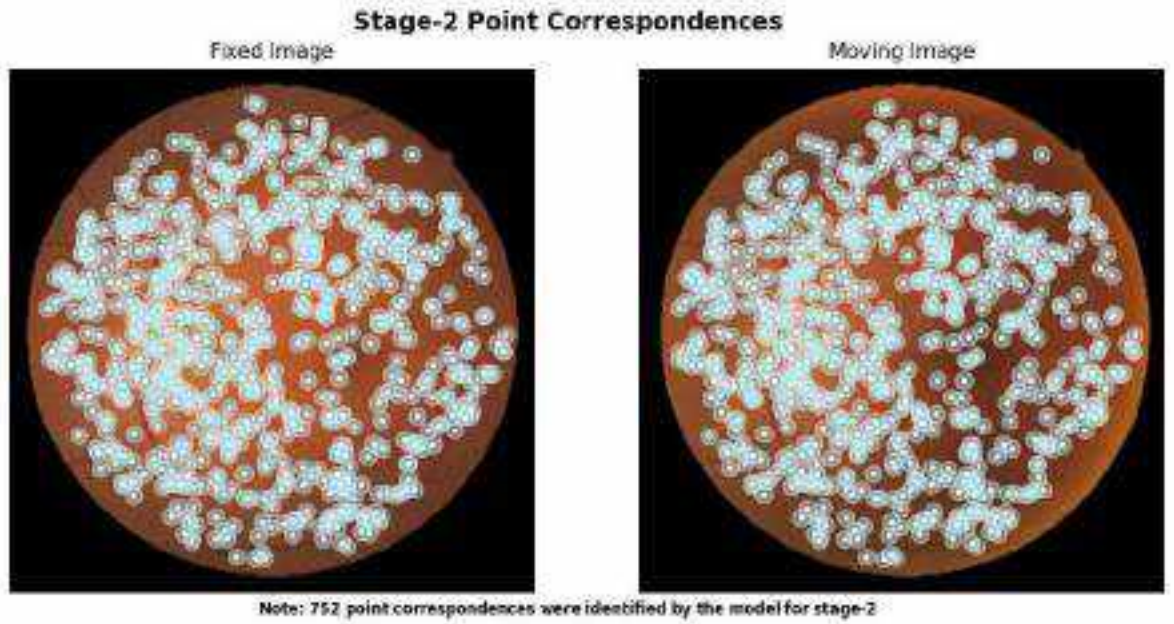

```
Mean Landmark Error for Case 20 Before Registration is 37.12218008965657 pixels
Mean Landmark Error for Case 20 After Registration is 2.054644537006522 pixels
Case 21
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S22_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S22_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



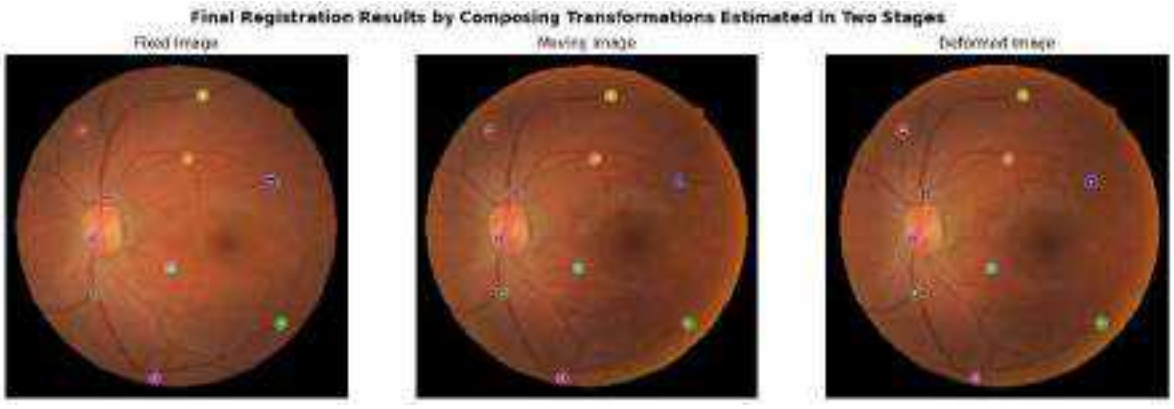

```
Homography Matrix:
[[ 9.98851499e-01  7.12773439e-04  7.52340672e+00]
 [-1.13451348e-03  9.99111271e-01  4.94131172e-01]
 [ 1.43845384e-06 -1.96278555e-06  1.00000000e+00]]
```



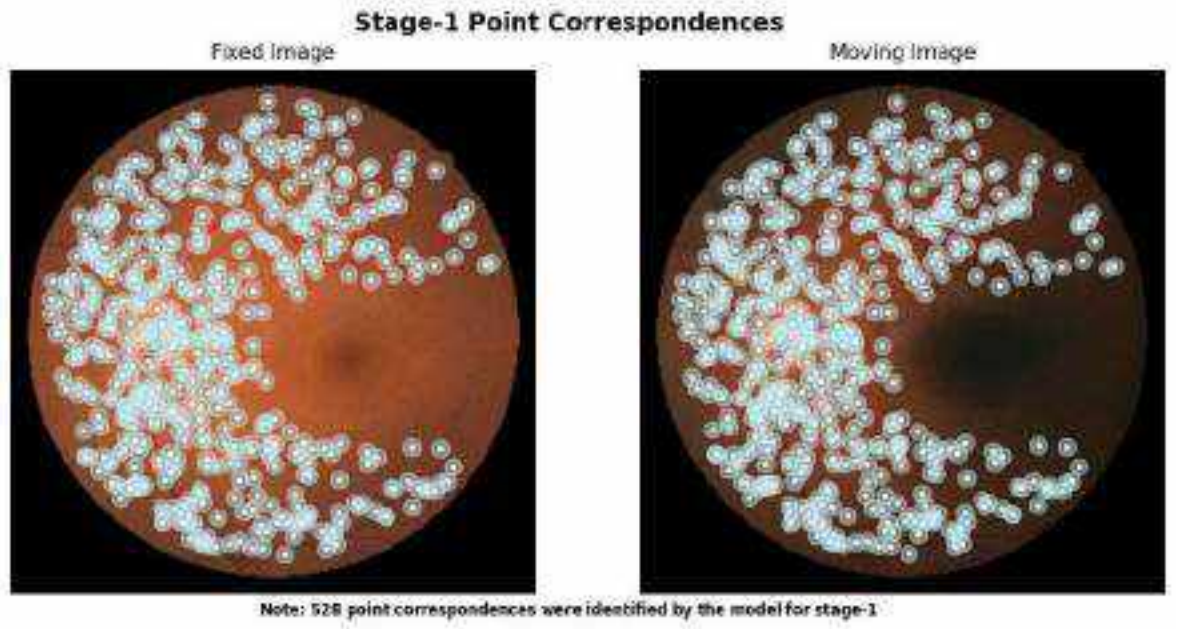

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



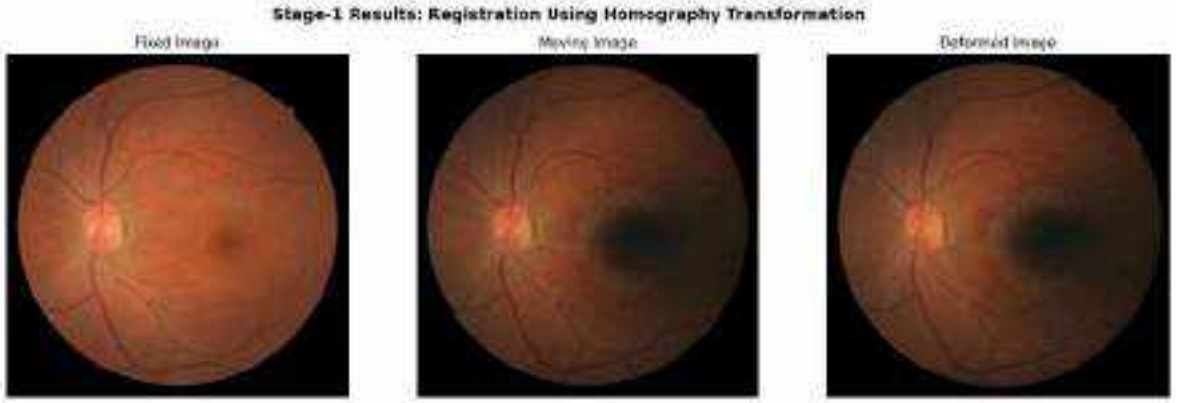
Note: 752 point correspondences were identified by the model for stage-2



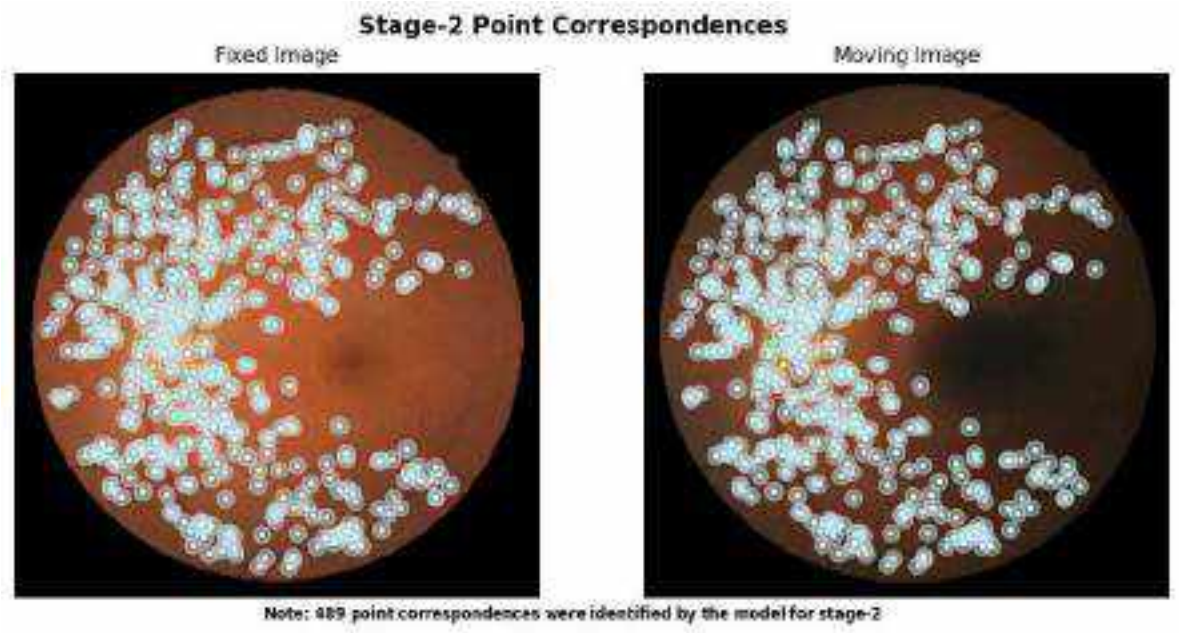
```
Mean Landmark Error for Case 21 Before Registration is 24.053298899313557 pixels
Mean Landmark Error for Case 21 After Registration is 1.6145369908851133 pixels
Case 22
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S23_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S23_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



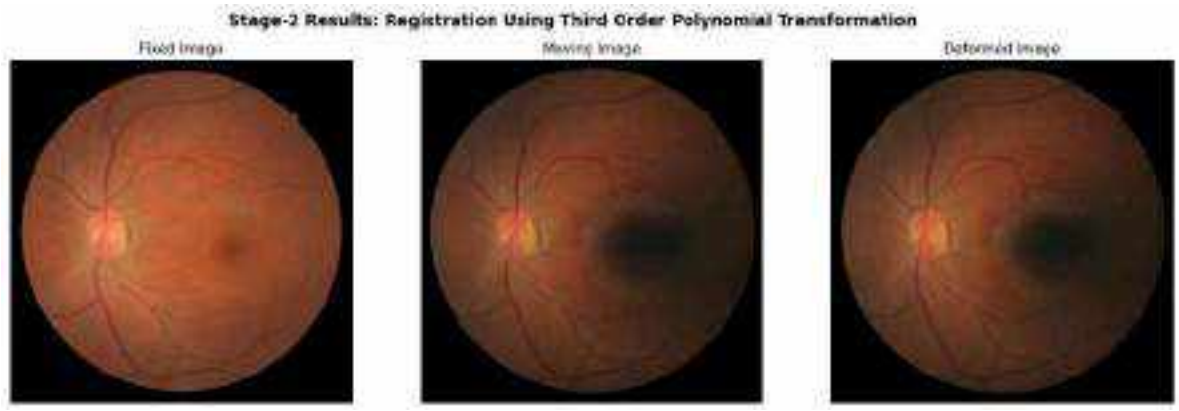

```
Homography Matrix:
[[ 1.00766465e+00  3.01818348e-03 -3.27652734e+00]
 [-4.82139241e-03  1.00923228e+00 -4.96251906e+00]
 [-4.54270319e-06 -5.36110671e-07  1.00000000e+00]]
```

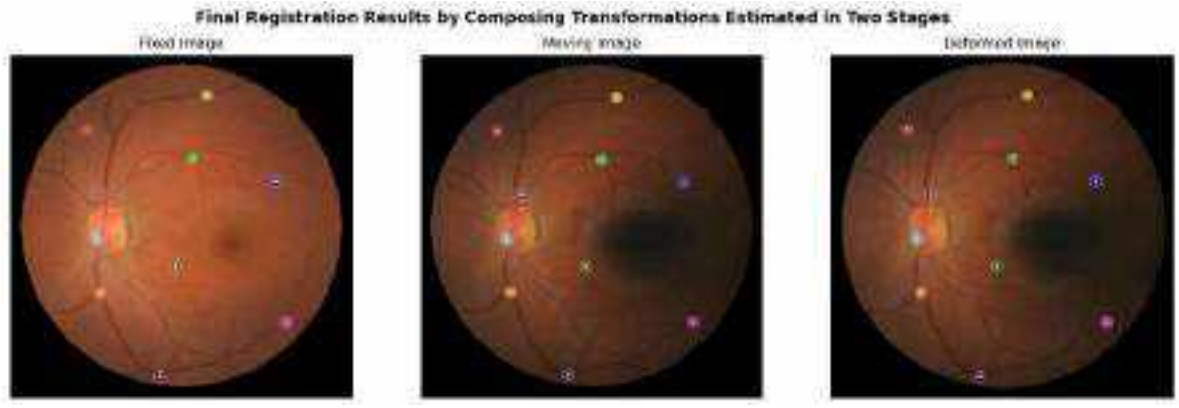



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

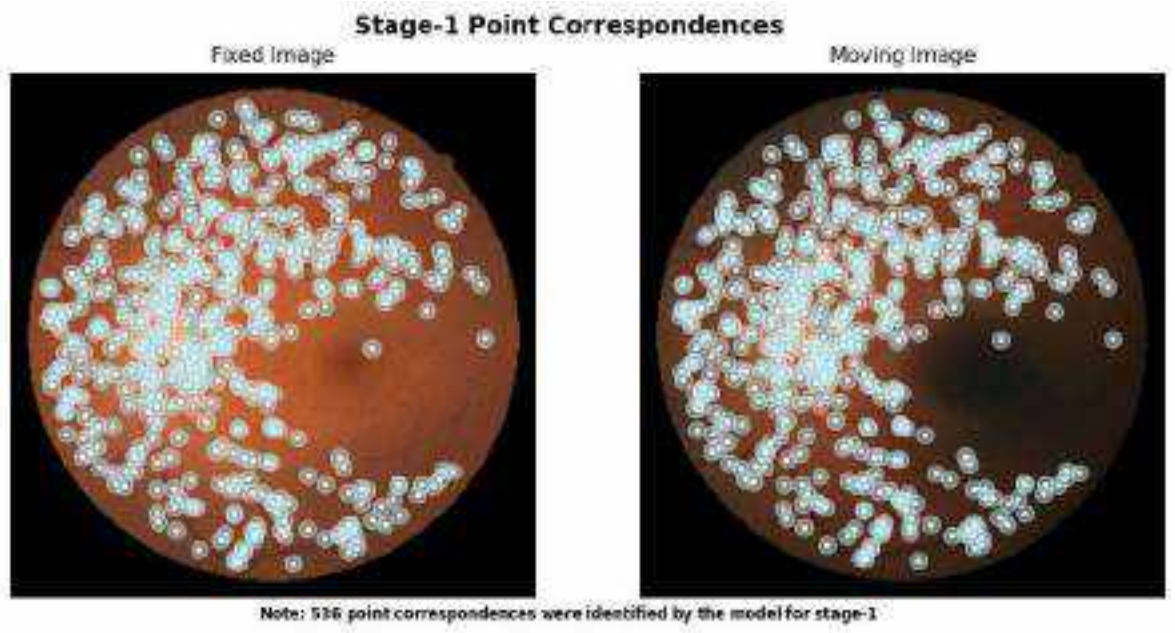


Note: 489 point correspondences were identified by the model for stage-2



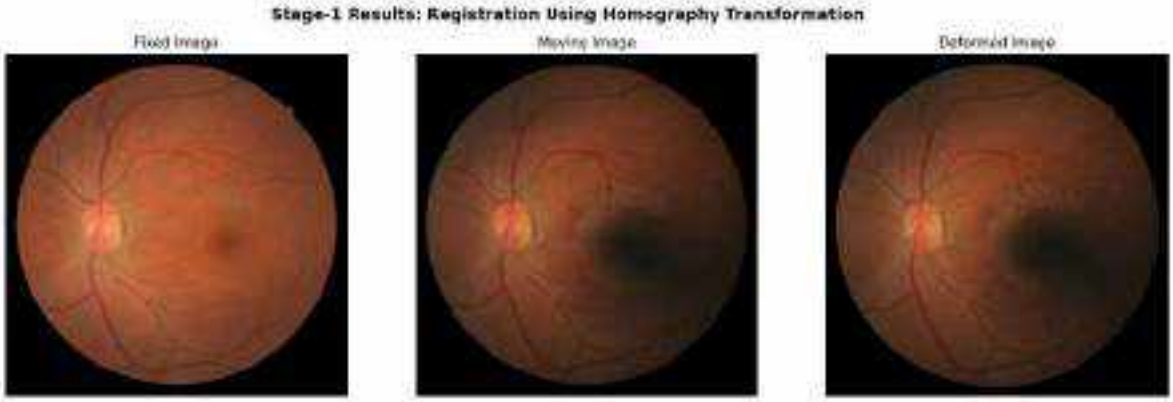



```
Mean Landmark Error for Case 22 Before Registration is 14.285985923547031 pixels
Mean Landmark Error for Case 22 After Registration is 1.6445940343073358 pixels
Case 23
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S24_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S24_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

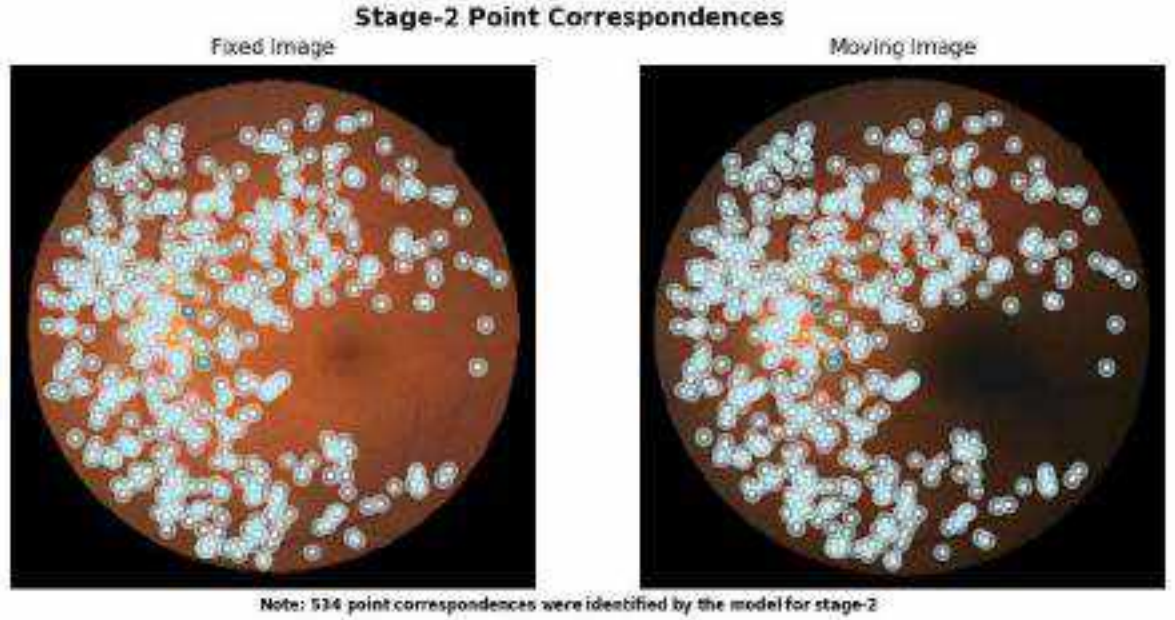


Note: 536 point correspondences were identified by the model for stage-1

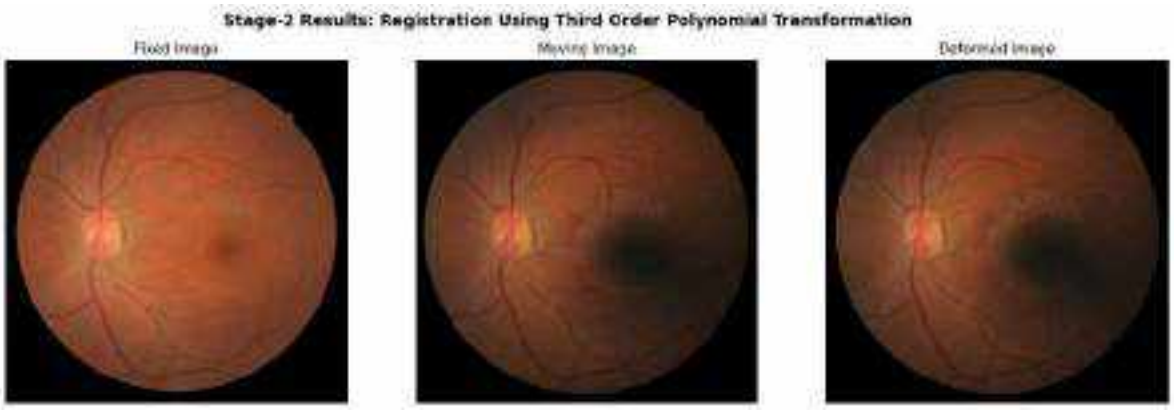
```
Homography Matrix:
[[ 1.01448829e+00  1.02506603e-02 -9.84239916e+00]
 [-9.91040569e-03  1.01126635e+00  4.96275498e-01]
 [ 4.60831343e-06 -3.33648643e-06  1.00000000e+00]]
```

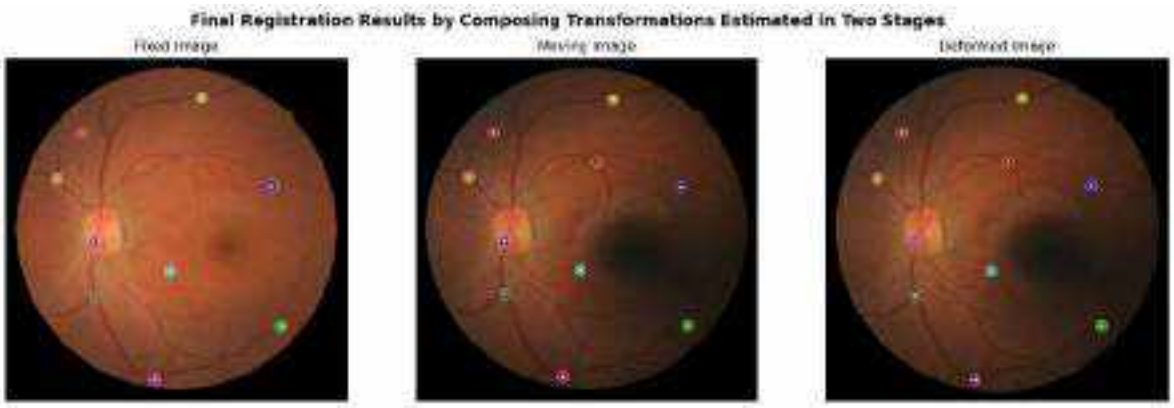



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

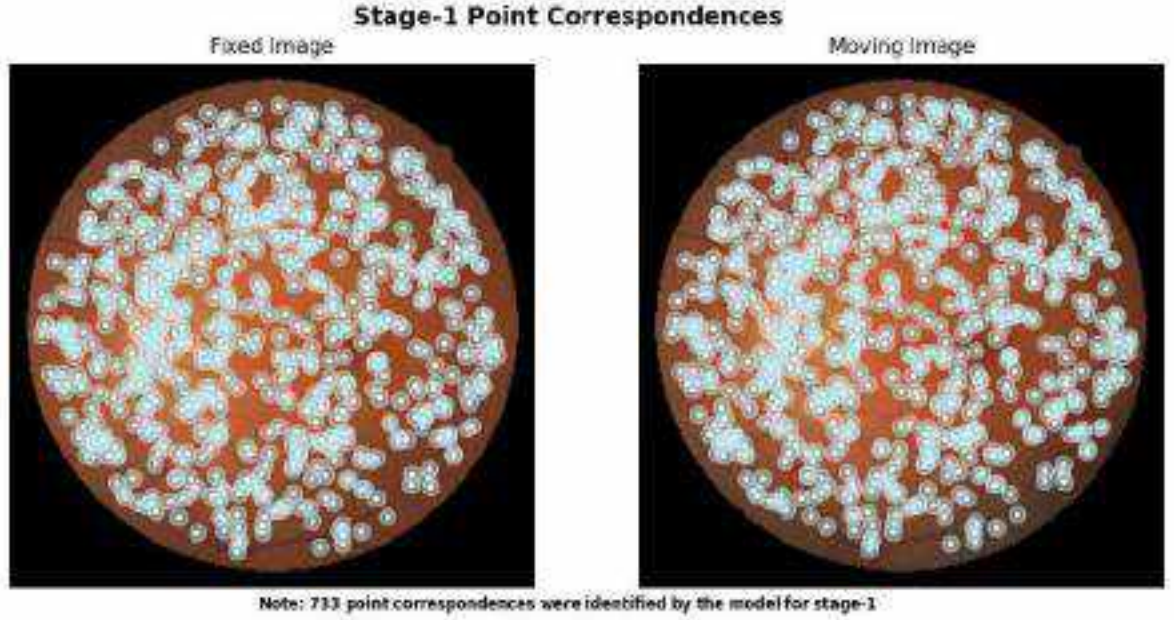


Note: 534 point correspondences were identified by the model for stage-2

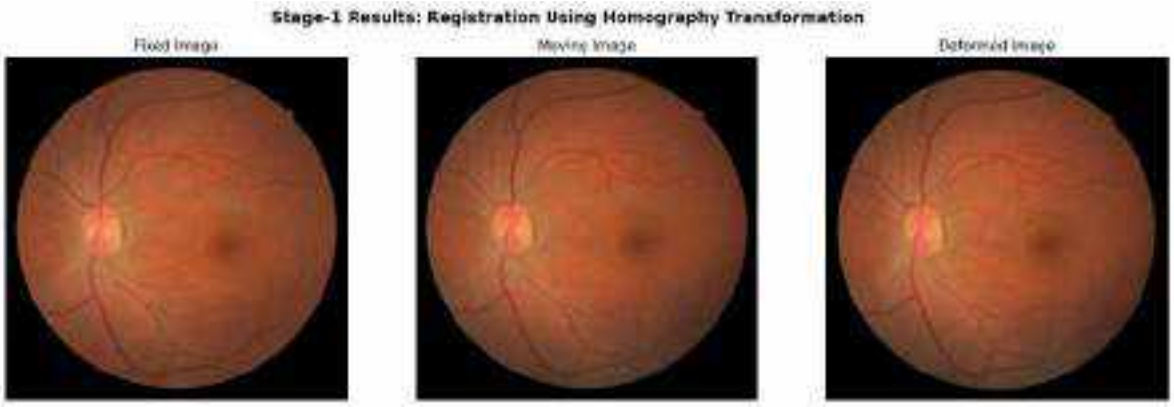




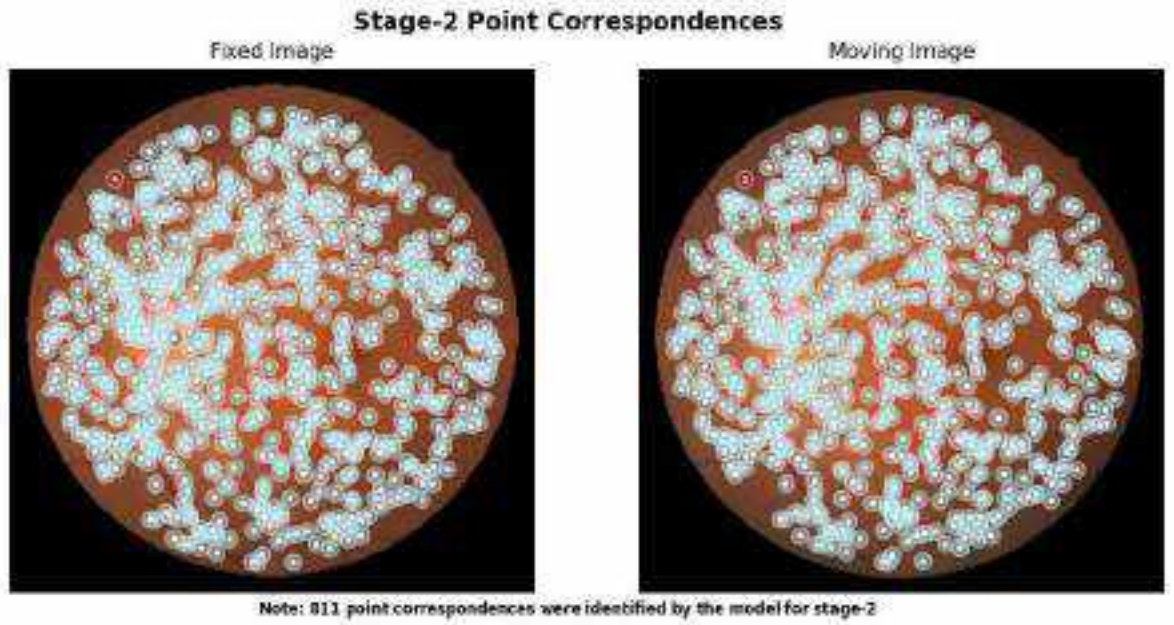

```
Mean Landmark Error for Case 23 Before Registration is 17.239695291533444 pixels
Mean Landmark Error for Case 23 After Registration is 2.080694803365767 pixels
Case 24
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S25_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S25_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



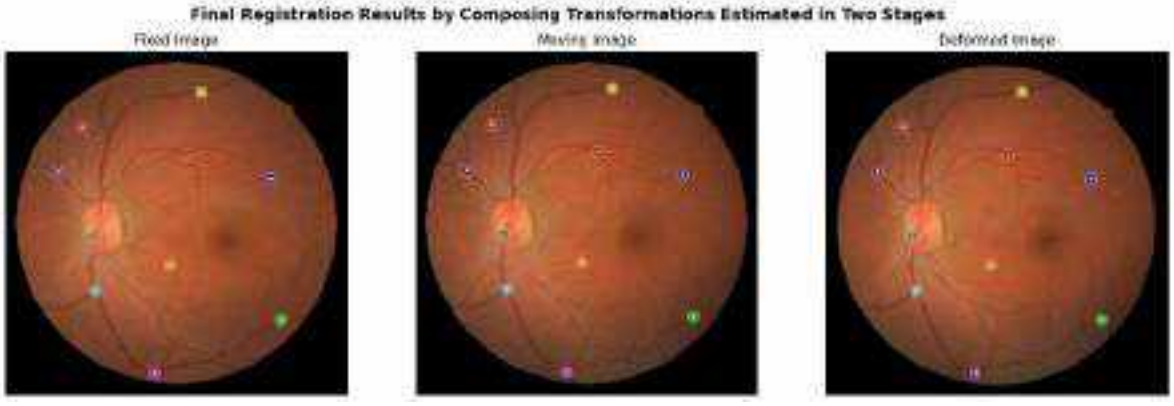

```
Homography Matrix:
[[ 9.99352501e-01 -3.45064108e-03  1.43617762e+00]
 [ 9.38134438e-03  1.00075688e+00  4.76988582e+00]
 [ 3.13215928e-06  6.13104277e-06  1.00000000e+00]]
```



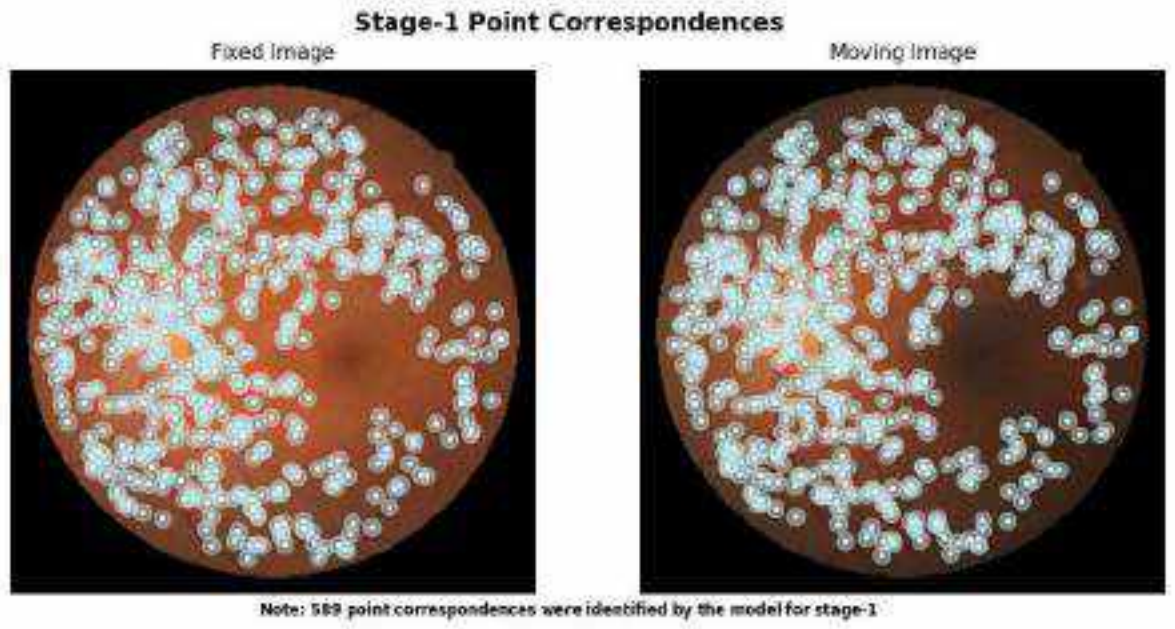

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



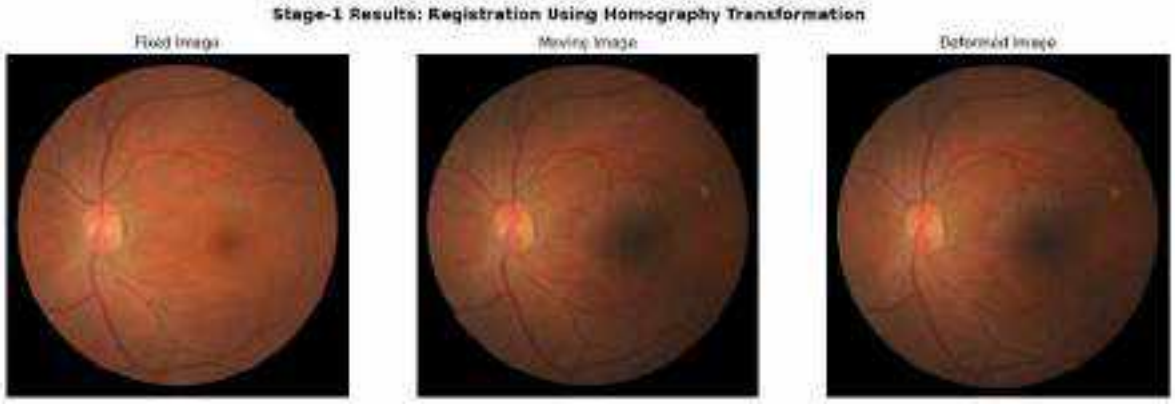
Note: 811 point correspondences were identified by the model for stage-2



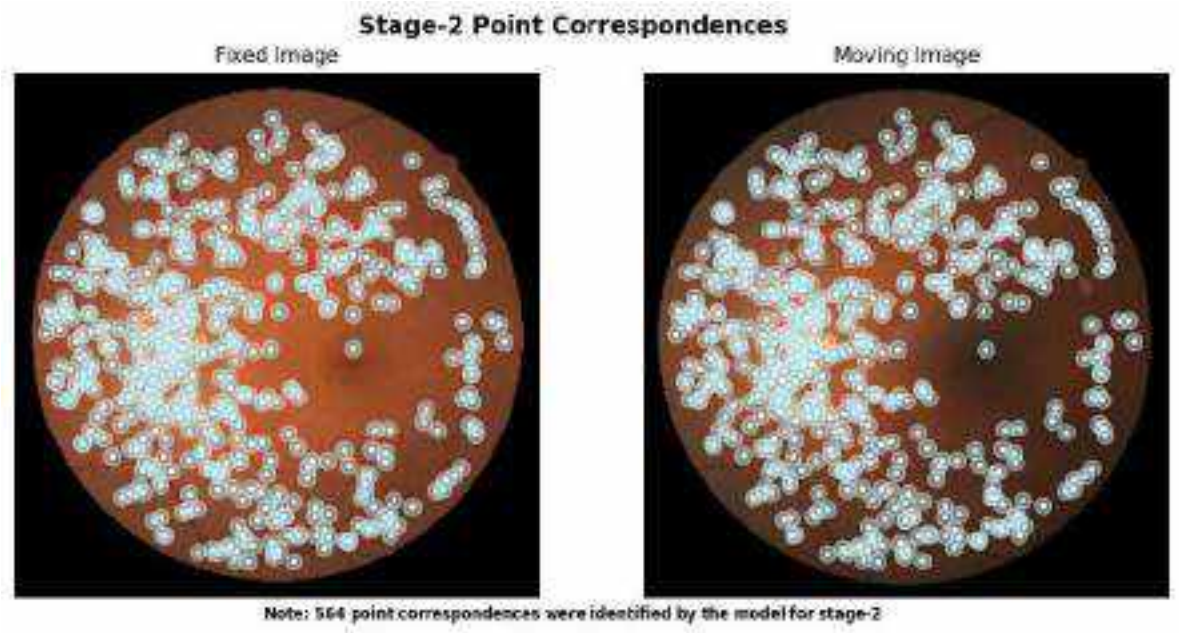
```
Mean Landmark Error for Case 24 Before Registration is 25.41908794622944 pixels
Mean Landmark Error for Case 24 After Registration is 1.4151712413925992 pixels
Case 25
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S26_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S26_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

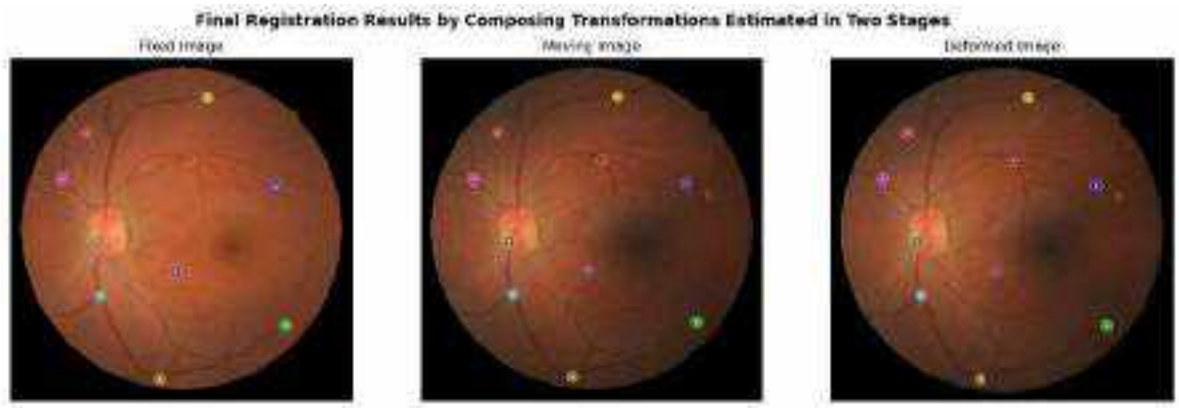



```
Homography Matrix:
[[ 1.00614016e+00 -9.91700792e-03  4.55245596e-02]
 [ 1.08550451e-02  1.00823486e+00 -4.33602365e+00]
 [ 1.50402759e-06  2.43018480e-07  1.00000000e+00]]
```

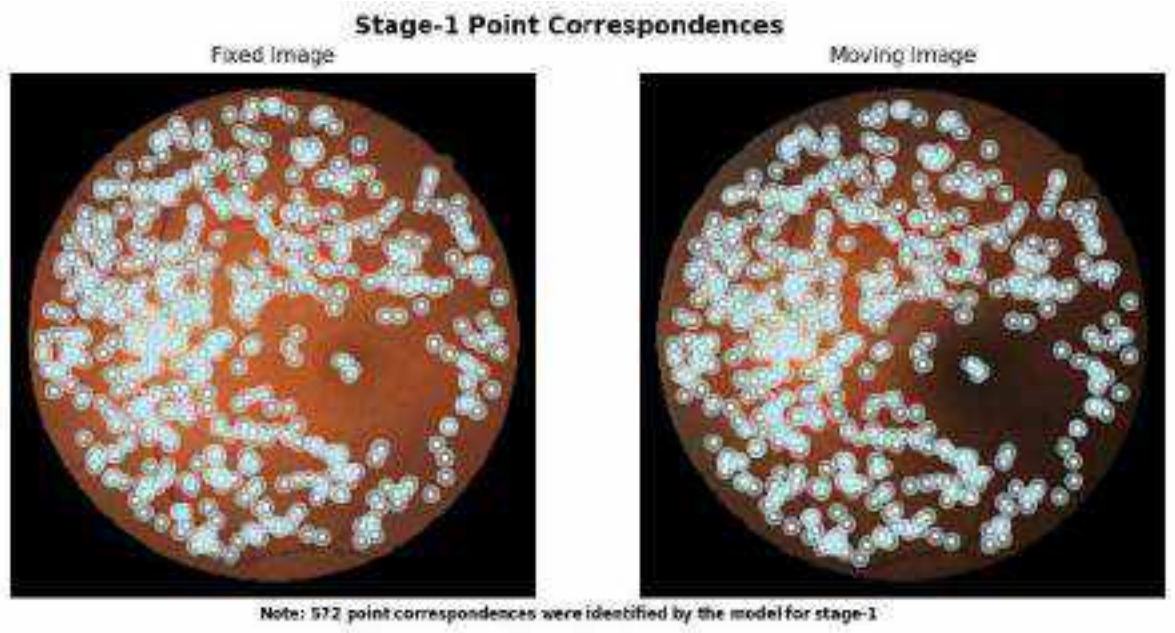



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



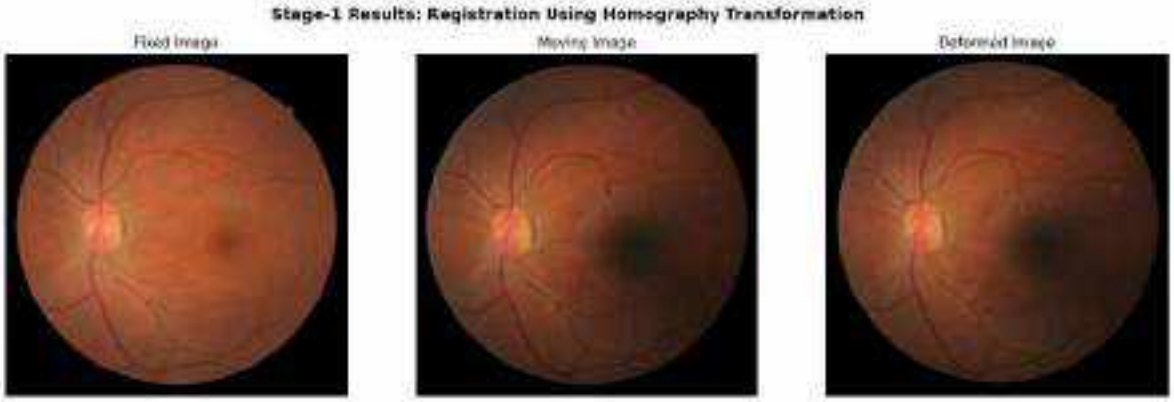



```
Mean Landmark Error for Case 25 Before Registration is 17.414954769611278 pixels
Mean Landmark Error for Case 25 After Registration is 1.6295964087415158 pixels
Case 26
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S27_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S27_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

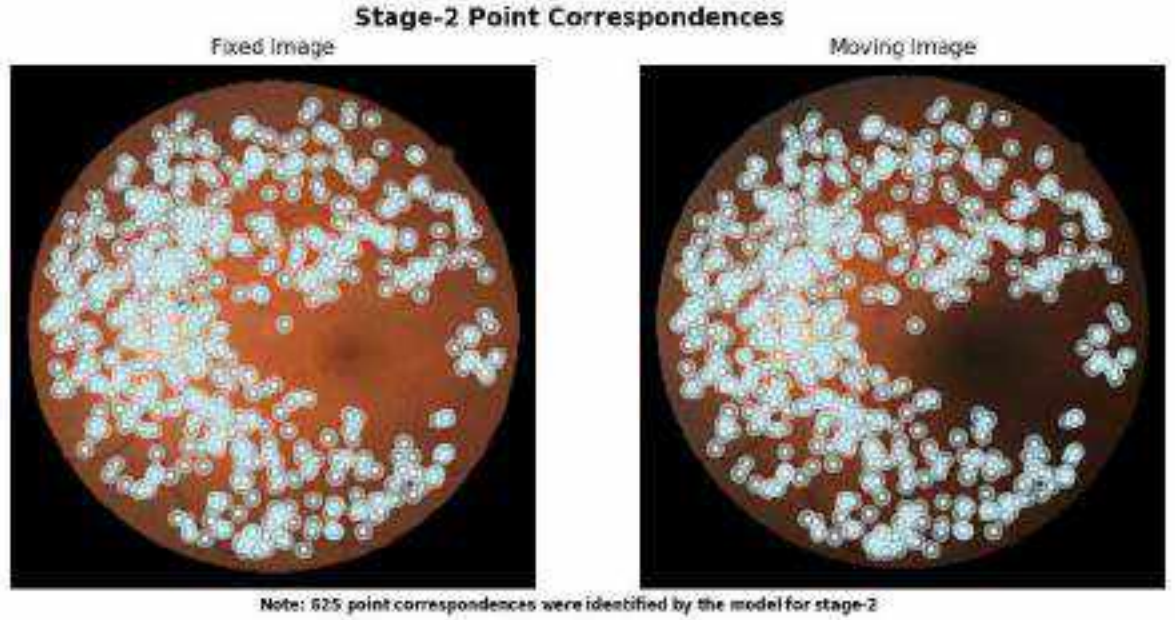


Note: 572 point correspondences were identified by the model for stage-1

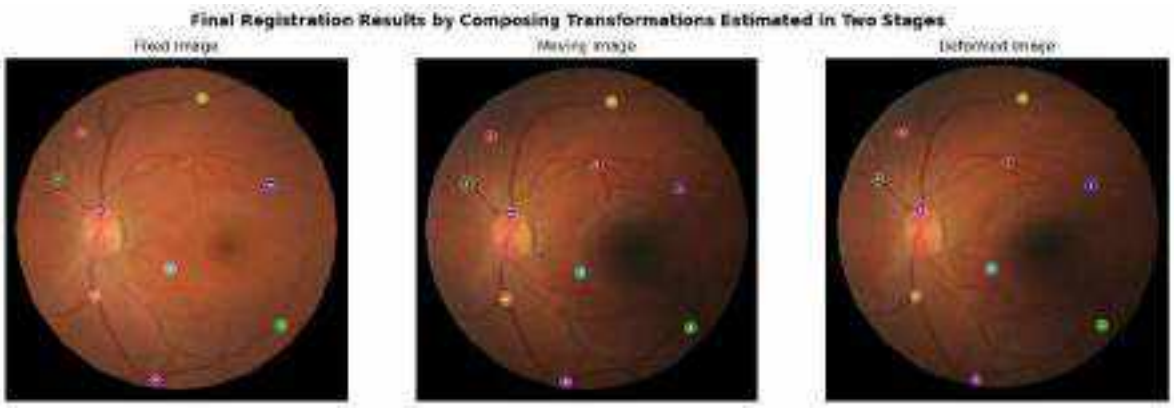
```
Homography Matrix:
[[ 9.99993454e-01 -7.35401588e-03  3.01148310e+00]
 [ 5.16354701e-03  9.98010057e-01 -1.21973534e+01]
 [-8.79129381e-07 -7.71733822e-06  1.00000000e+00]]
```



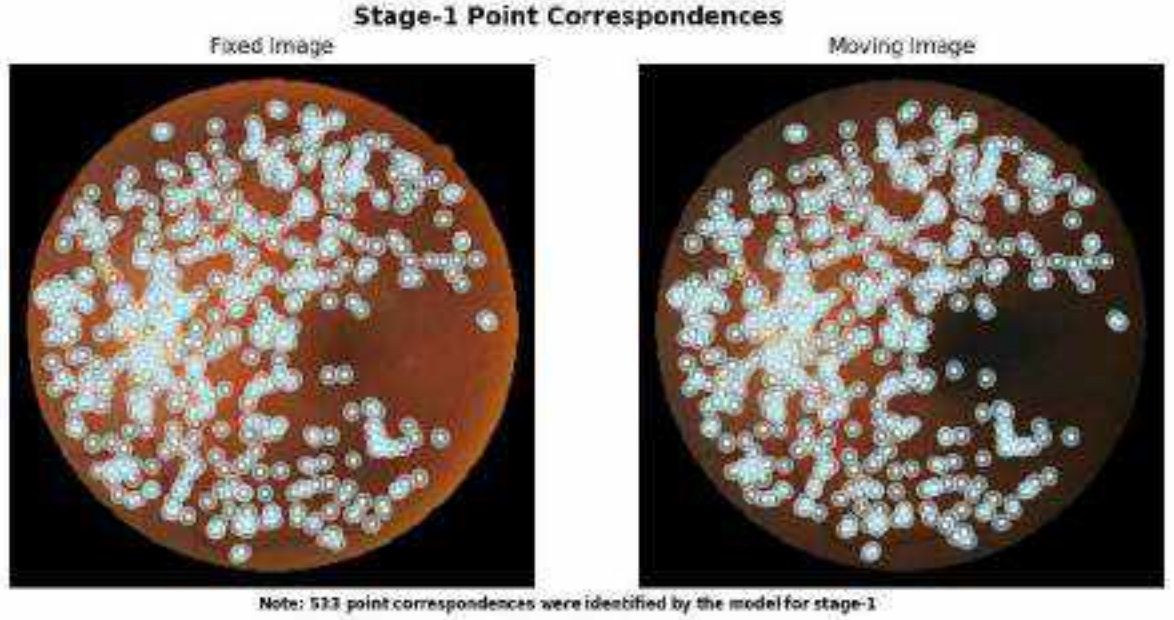

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



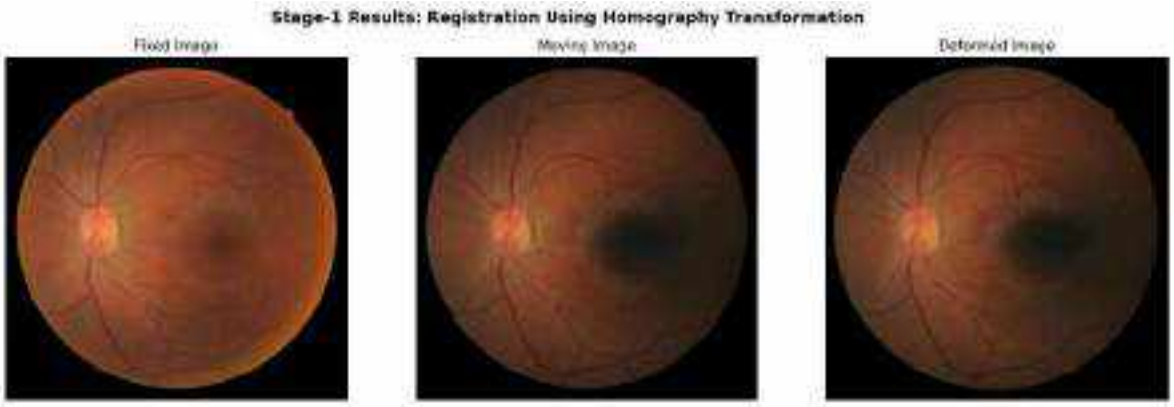
Note: 625 point correspondences were identified by the model for stage-2



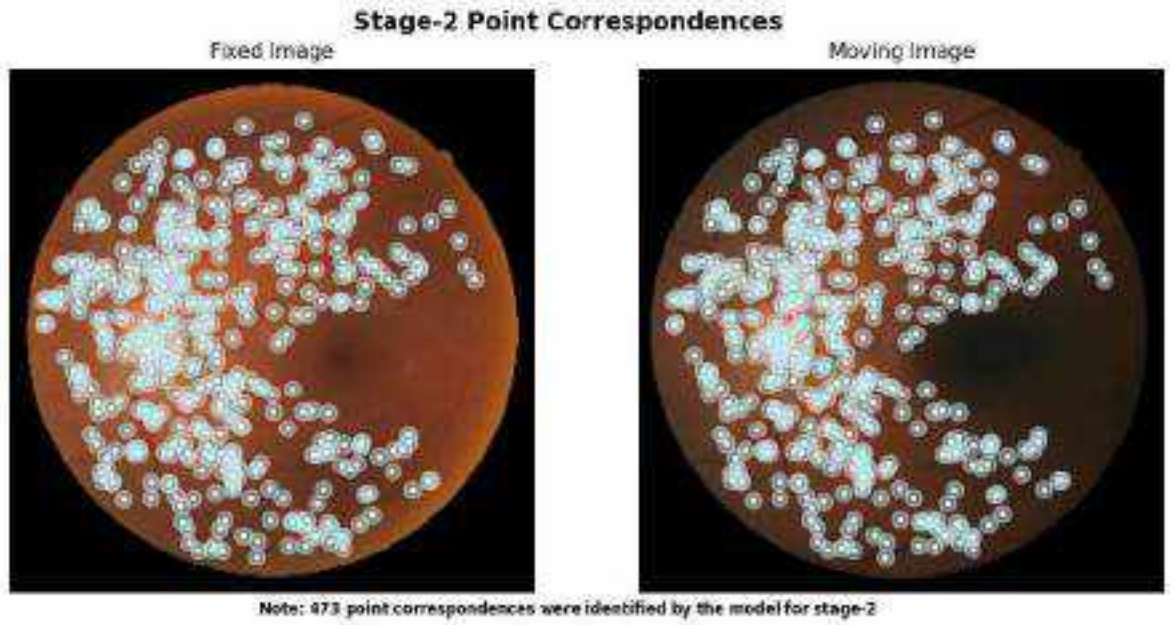

```
Mean Landmark Error for Case 26 Before Registration is 29.189946356744617 pixels
Mean Landmark Error for Case 26 After Registration is 1.4668631471666123 pixels
Case 27
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S28_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S28_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



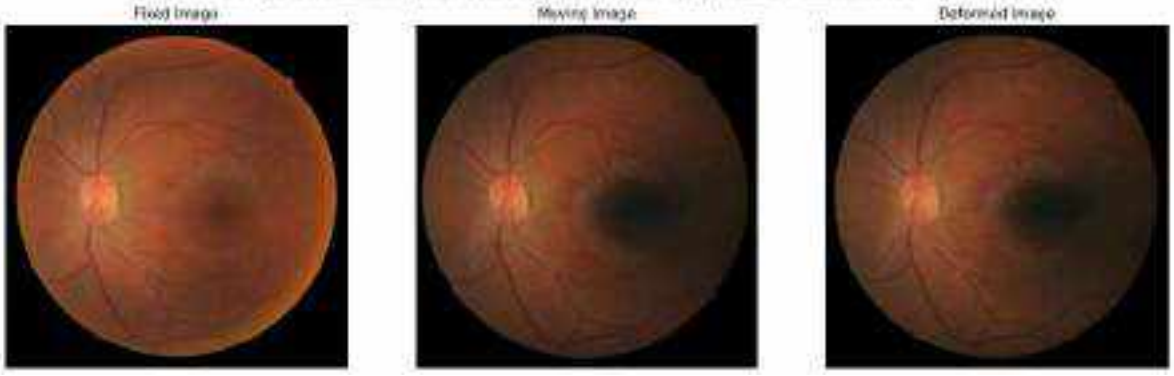

```
Homography Matrix:
[[ 1.00948581e+00  6.96988938e-04 -9.90510307e+00]
 [-2.19074982e-03  1.00784237e+00 -5.44392337e+00]
 [-3.61519169e-06 -2.62047055e-06  1.00000000e+00]]
```

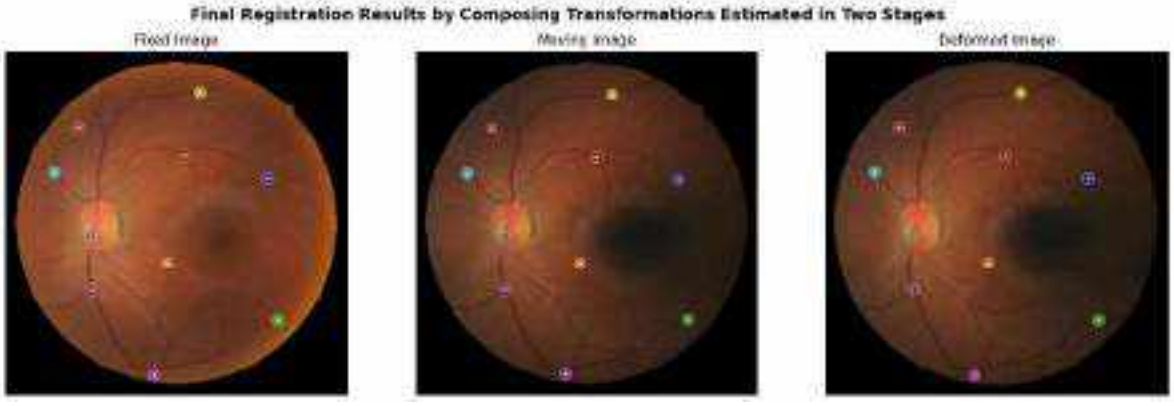



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

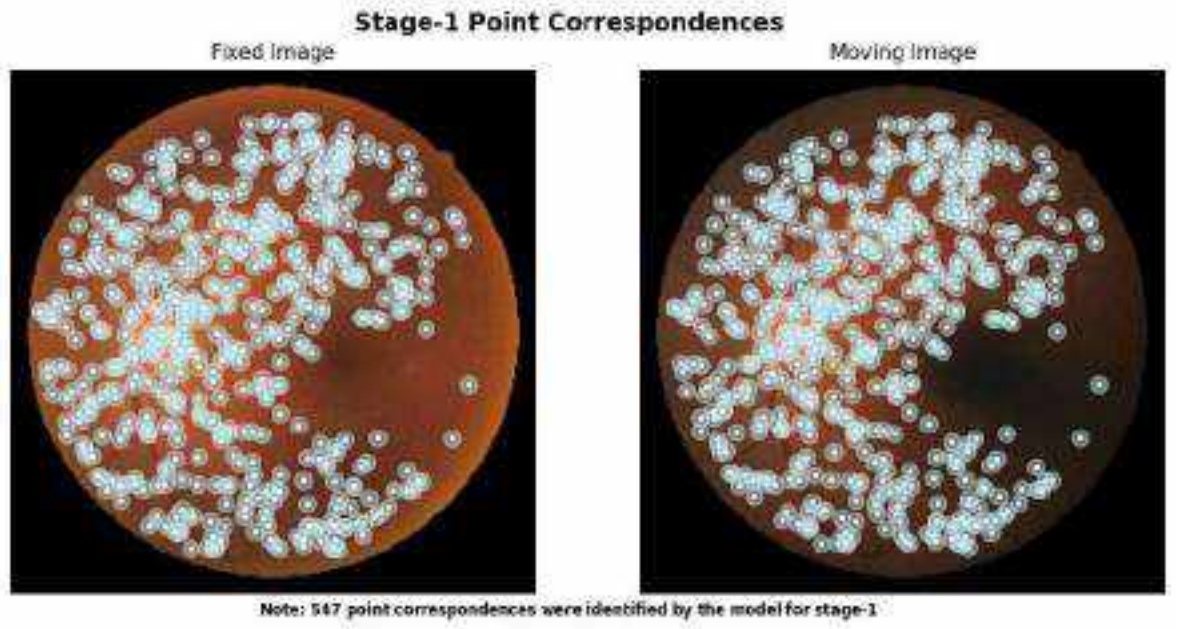


Note: 473 point correspondences were identified by the model for stage-2

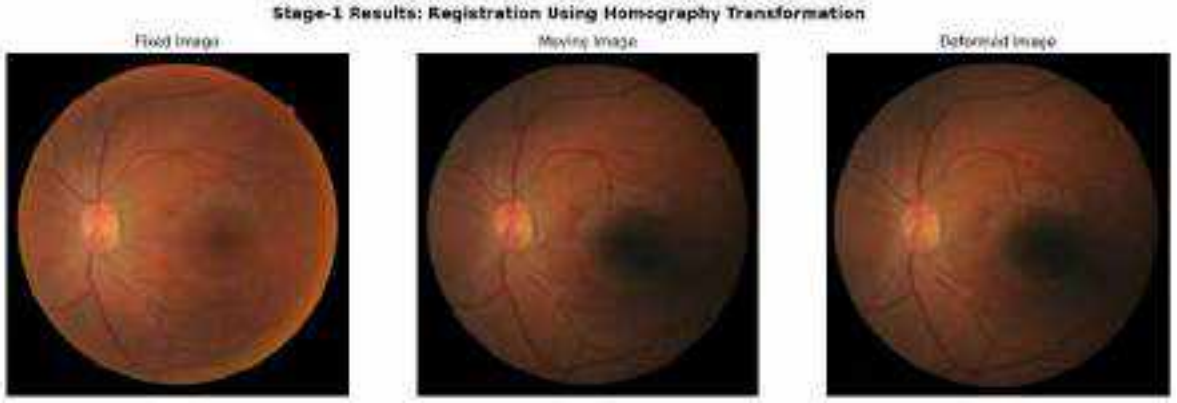




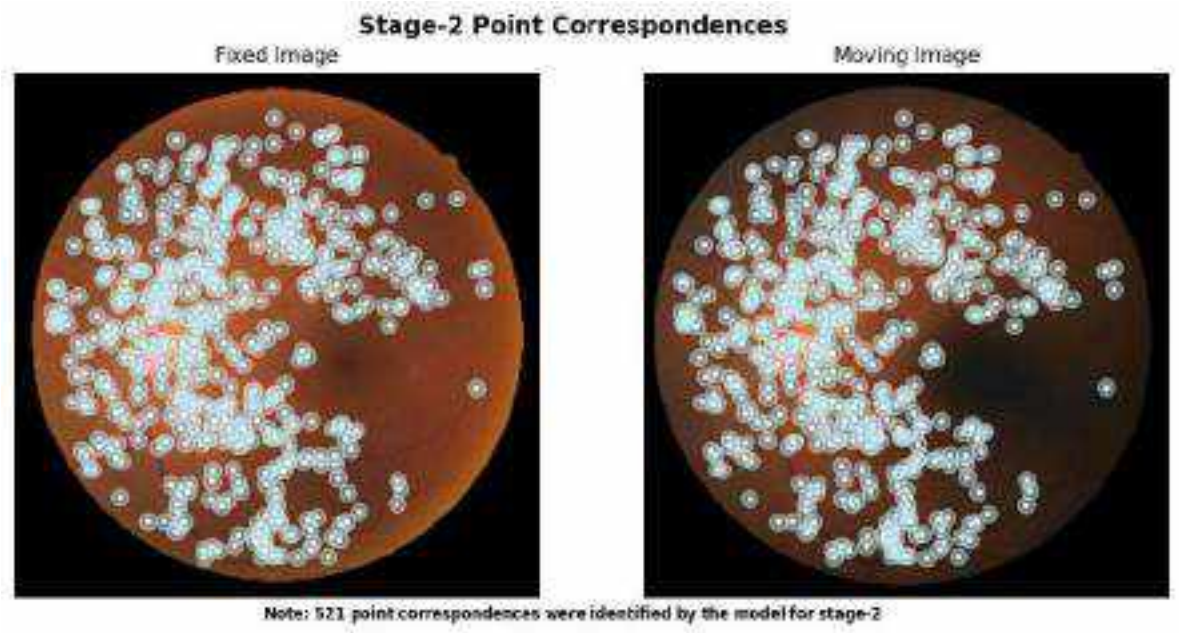
```
Mean Landmark Error for Case 27 Before Registration is 19.560420313565561 pixels
Mean Landmark Error for Case 27 After Registration is 1.5337309717987002 pixels
Case 28
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S29_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S29_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

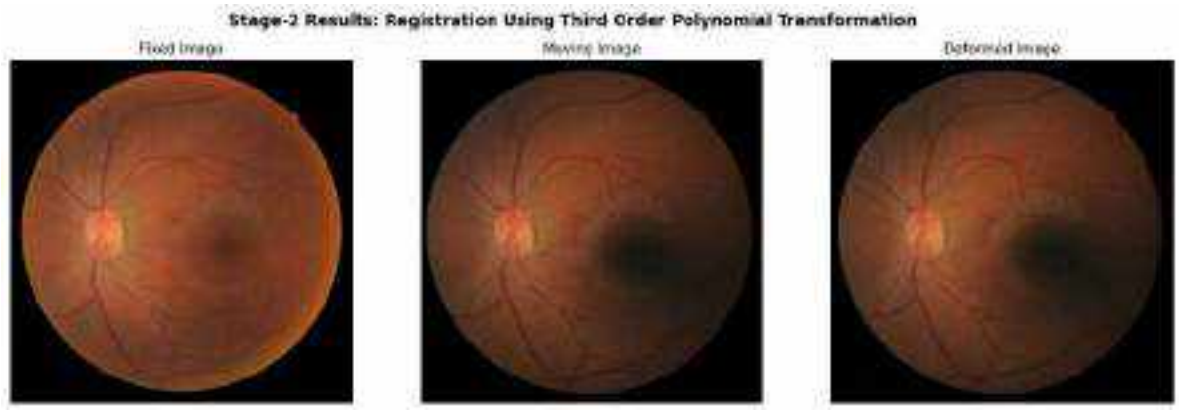


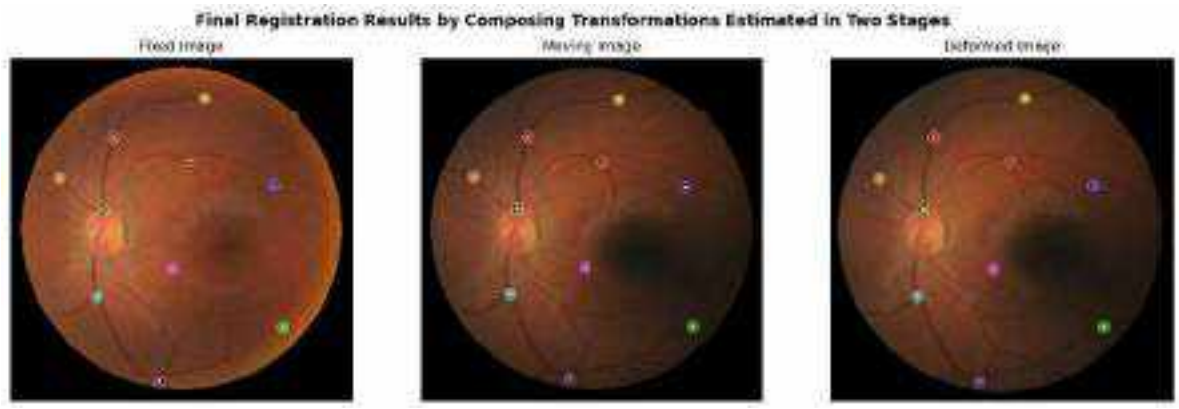

```
Homography Matrix:
[[ 1.01512216e+00  1.22495097e-02 -1.86330610e+01]
 [-8.97808955e-03  1.01244947e+00 -7.17047286e-02]
 [ 3.66658409e-07 -5.47682107e-07  1.00000000e+00]]
```



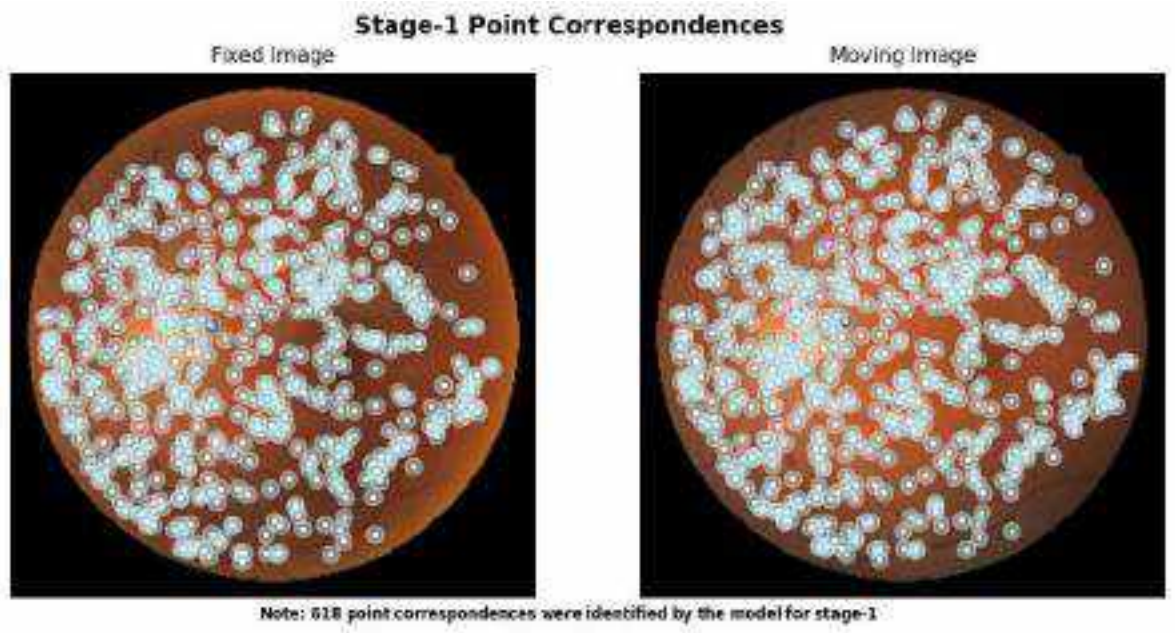

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```







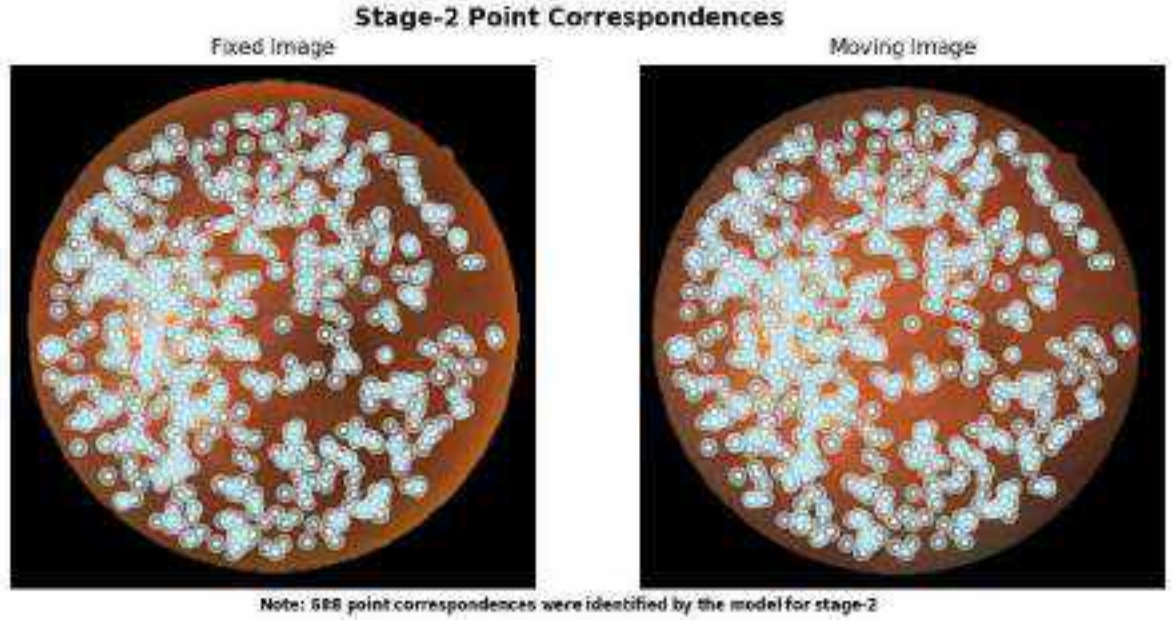

```
Mean Landmark Error for Case 28 Before Registration is 26.55489517738834 pixels
Mean Landmark Error for Case 28 After Registration is 1.9278216471668852 pixels
Case 29
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S30_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S30_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



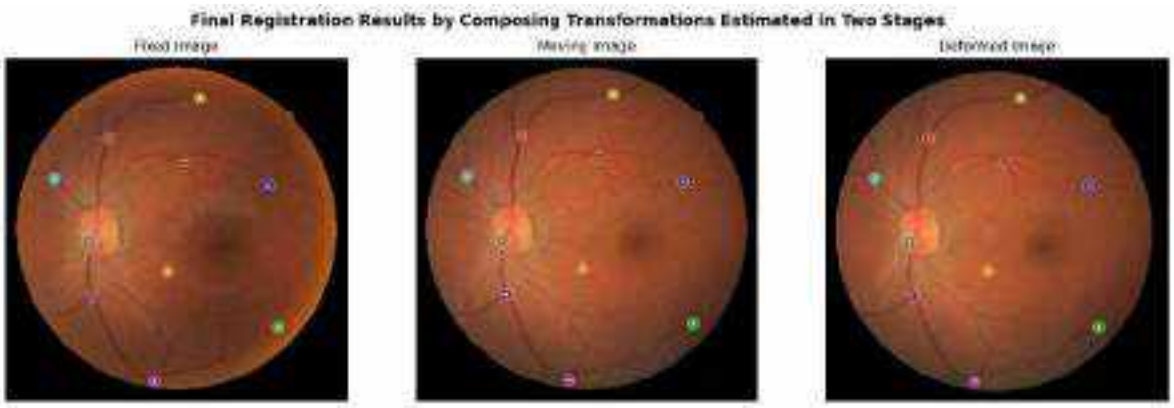
Note: 618 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.89625191e-01 -8.23681199e-03 -2.77615984e+00]
 [ 7.55124955e-03  9.92291192e-01  7.14215076e+00]
 [-4.95275818e-06  2.02658583e-06  1.00000000e+00]]
```

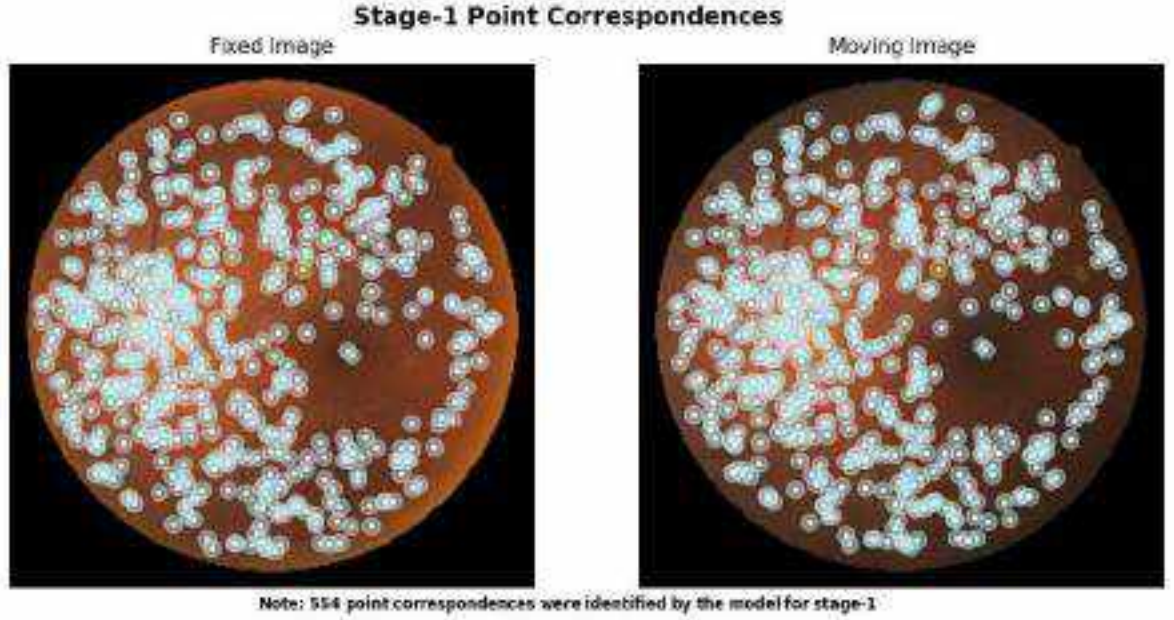

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



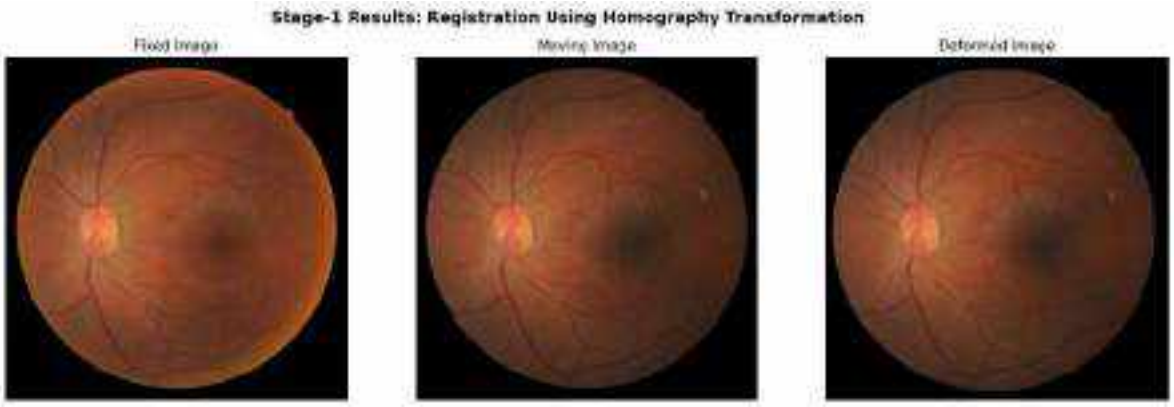
Note: 588 point correspondences were identified by the model for stage-2



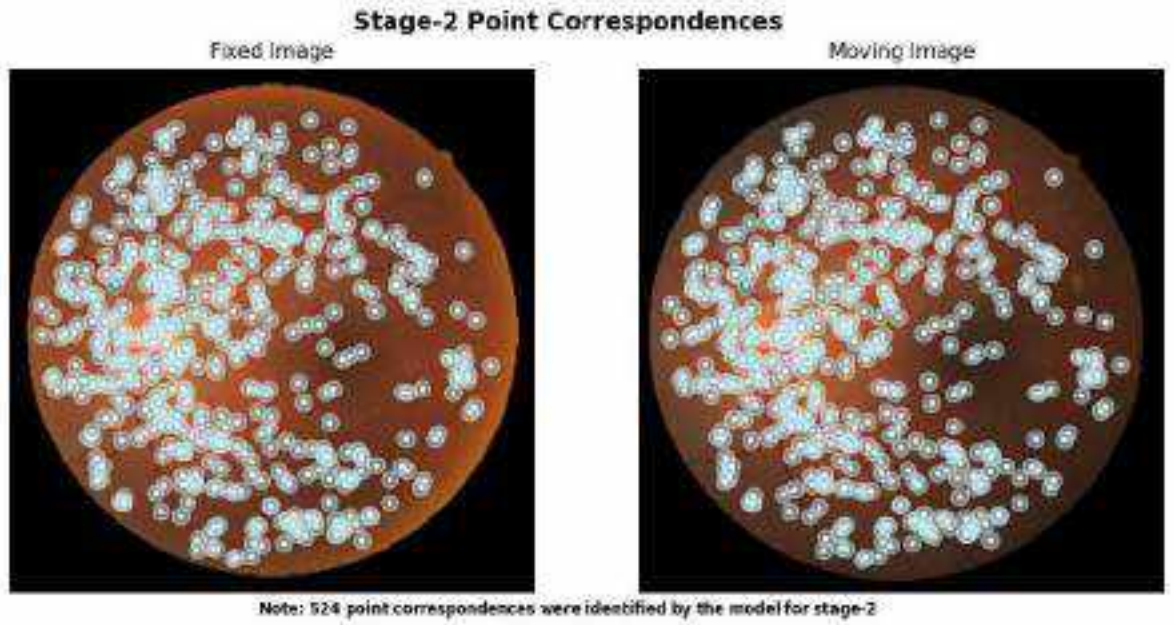

```
Mean Landmark Error for Case 29 Before Registration is 40.787678420943886 pixels
Mean Landmark Error for Case 29 After Registration is 1.39306959008117 pixels
Case 30
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S31_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S31_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



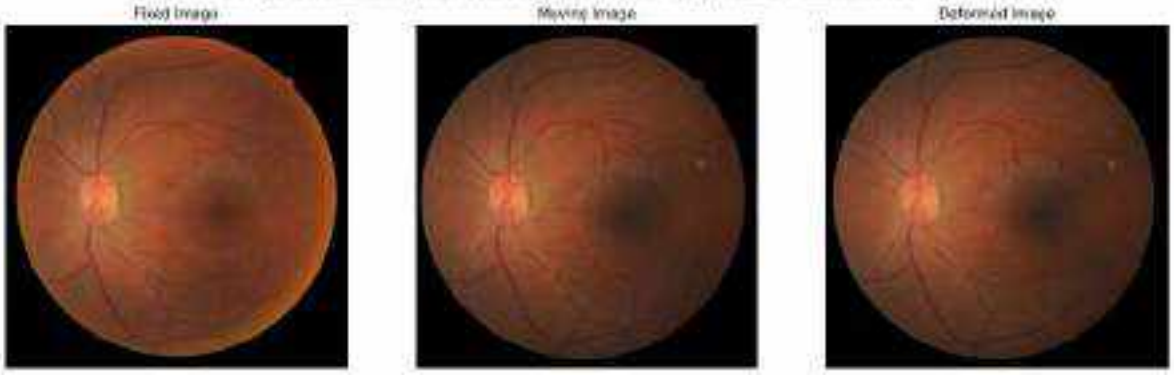

```
Homography Matrix:
[[ 9.99611775e-01 -1.43712884e-02 -5.99571692e+00]
 [ 7.89998037e-03  9.98940087e-01 -1.43589111e+00]
 [-6.77652473e-06 -4.78406817e-06  1.00000000e+00]]
```

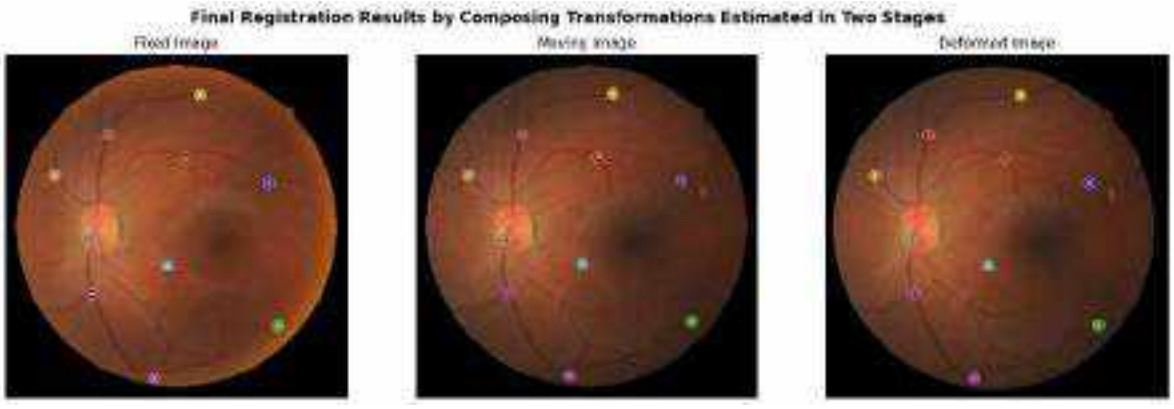



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

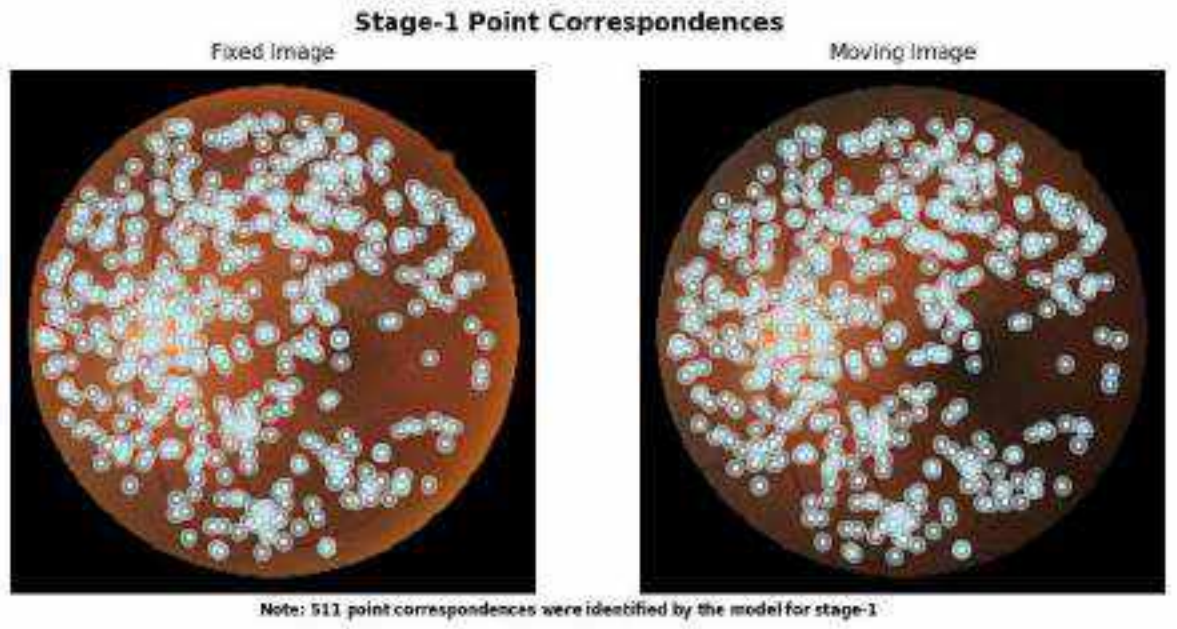


Note: 524 point correspondences were identified by the model for stage-2

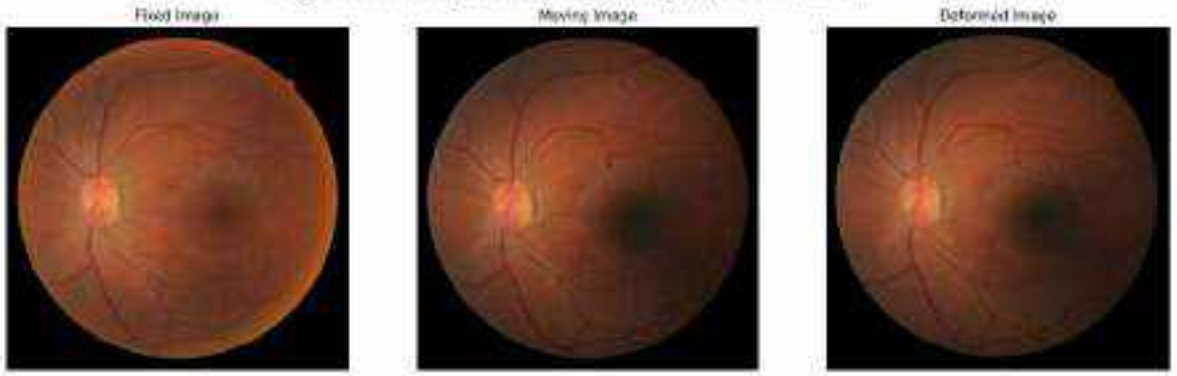




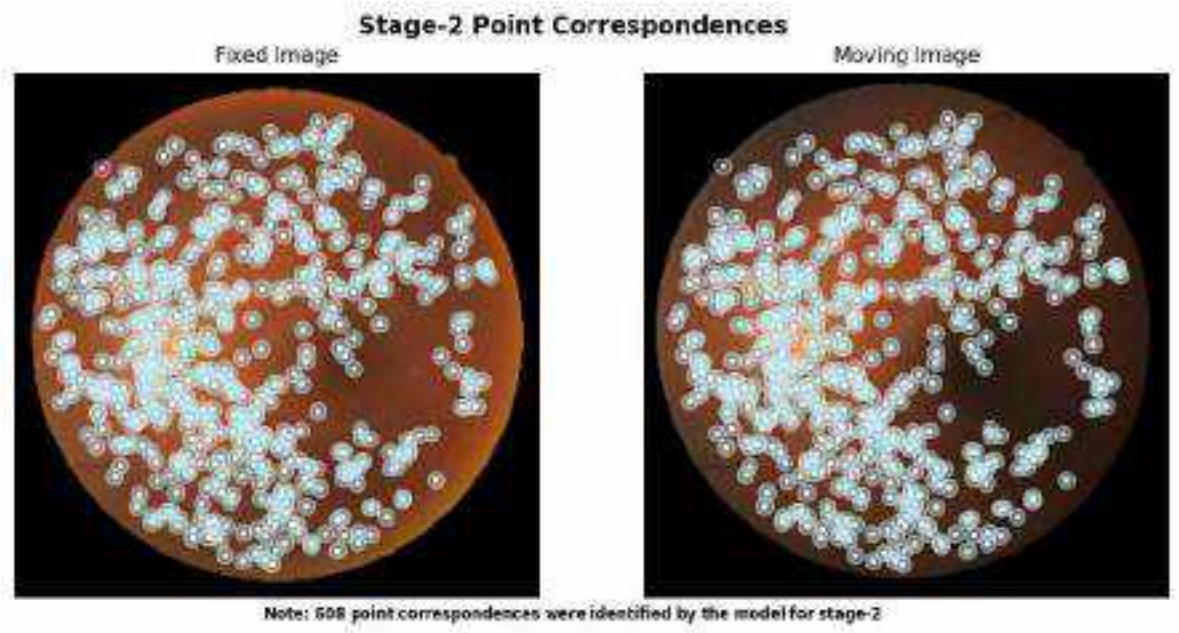
```
Mean Landmark Error for Case 30 Before Registration is 36.36294476139524 pixels
Mean Landmark Error for Case 30 After Registration is 1.560948390032551 pixels
Case 31
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S32_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S32_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

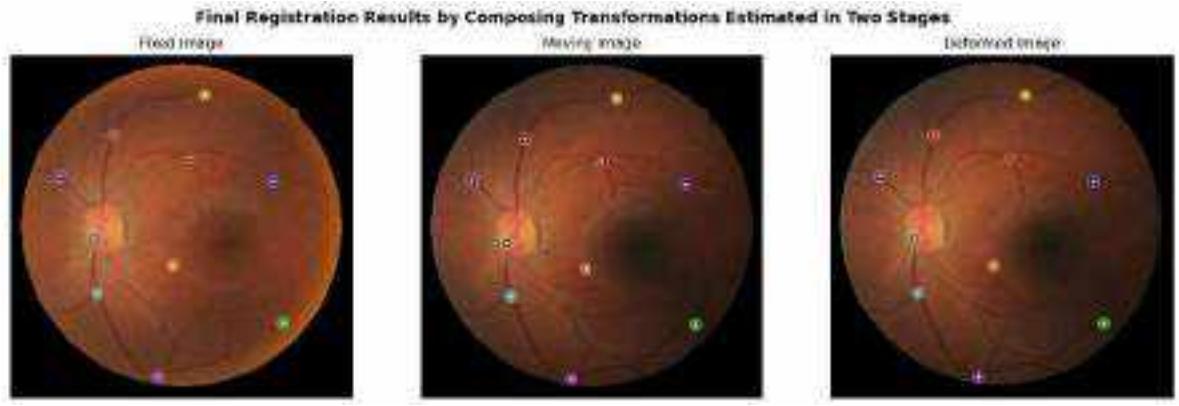



```
Homography Matrix:
[[ 1.00135889e+00 -1.10910943e-02 -2.98948216e+00]
 [ 8.79029806e-03  9.98123580e-01 -1.39921989e+01]
 [-1.51298212e-06 -8.82777349e-06  1.00000000e+00]]
```

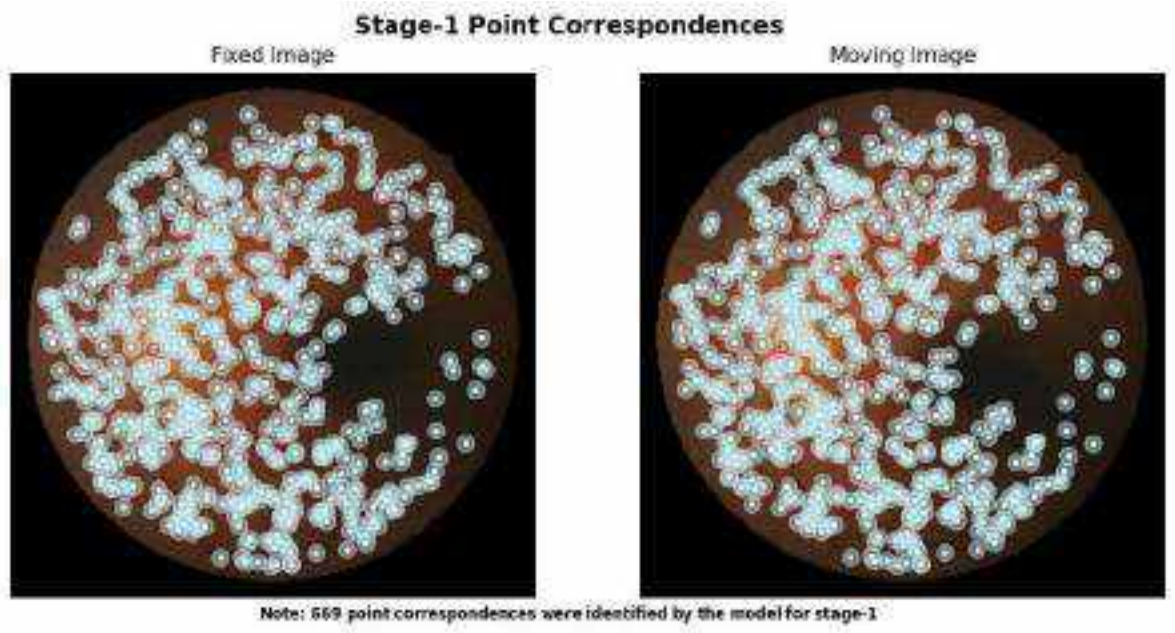



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





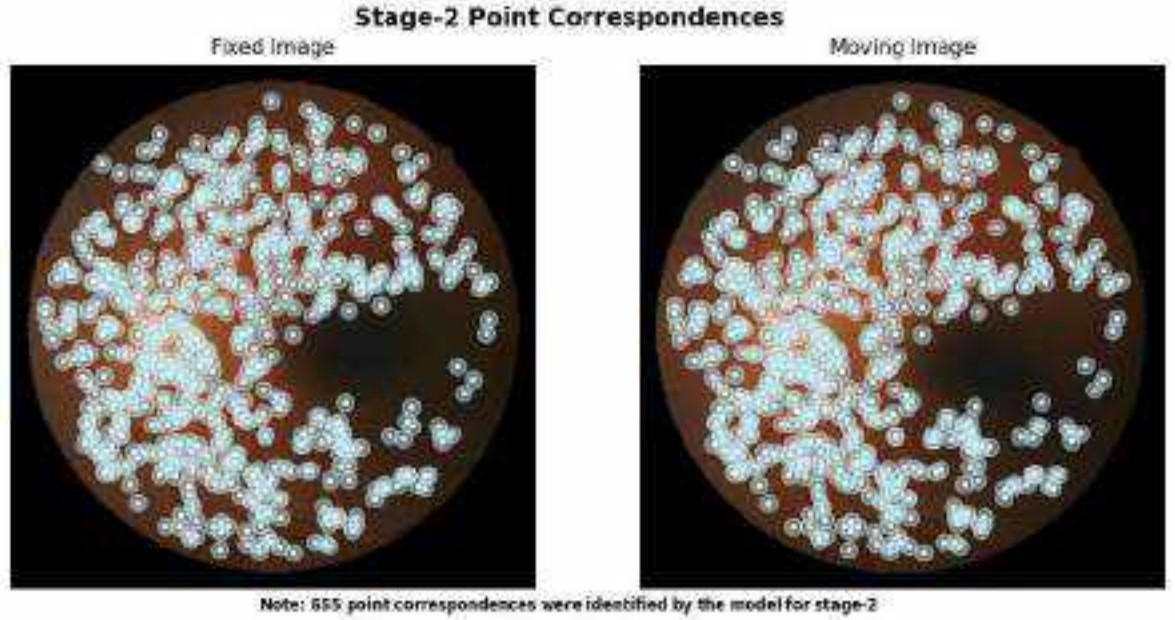

```
Mean Landmark Error for Case 31 Before Registration is 34.736813580073669 pixels
Mean Landmark Error for Case 31 After Registration is 1.9436864285088333 pixels
Case 32
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S33_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S33_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



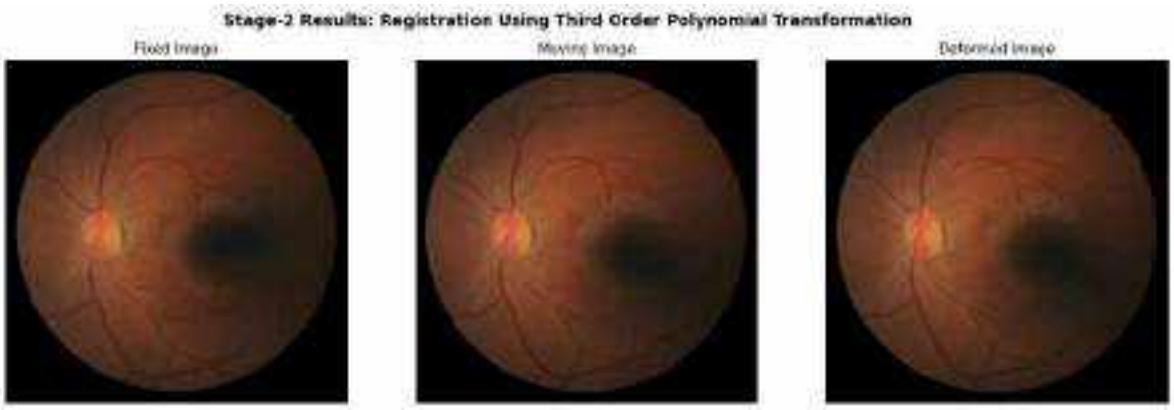
Note: 669 point correspondences were identified by the model for stage-1

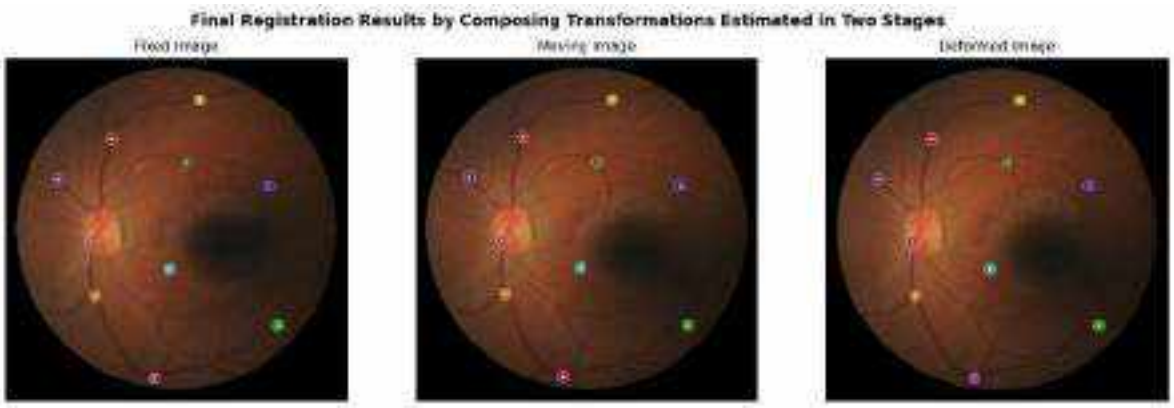
```
Homography Matrix:
[[ 1.00072417e+00  8.58461963e-03 -5.35826895e+00]
 [-1.00964197e-02  1.00189241e+00  7.00590129e+00]
 [ 1.40151927e-06  1.57953992e-06  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

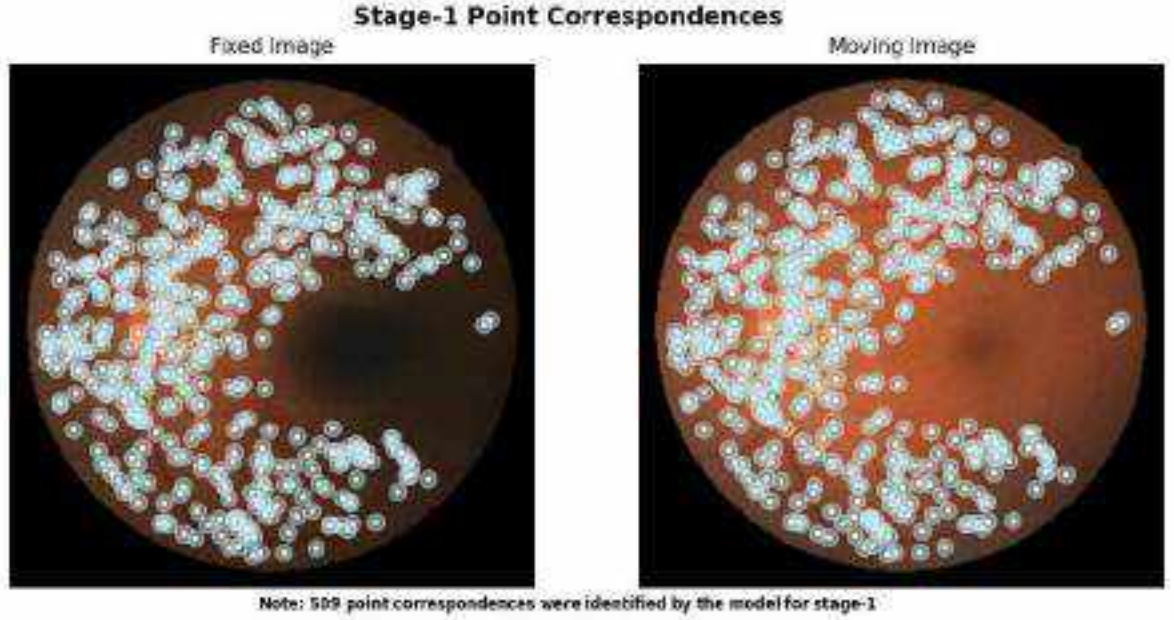


Note: 655 point correspondences were identified by the model for stage-2

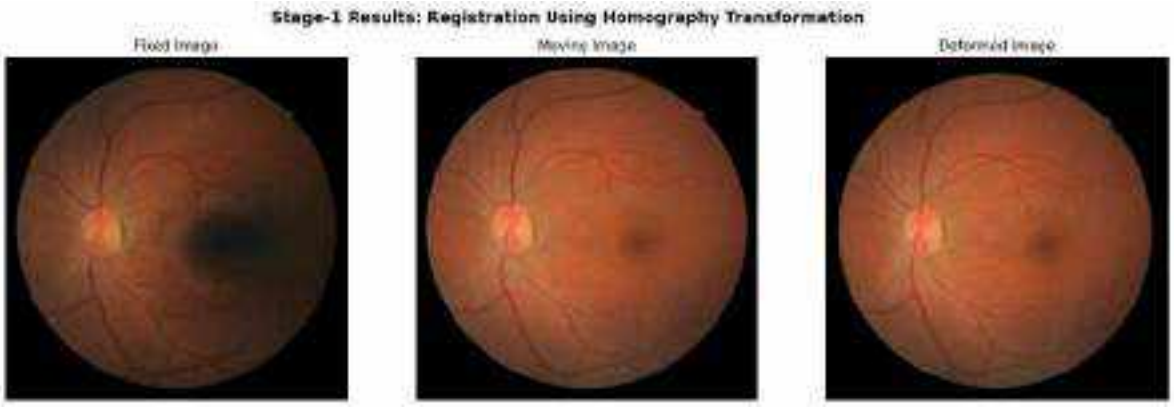




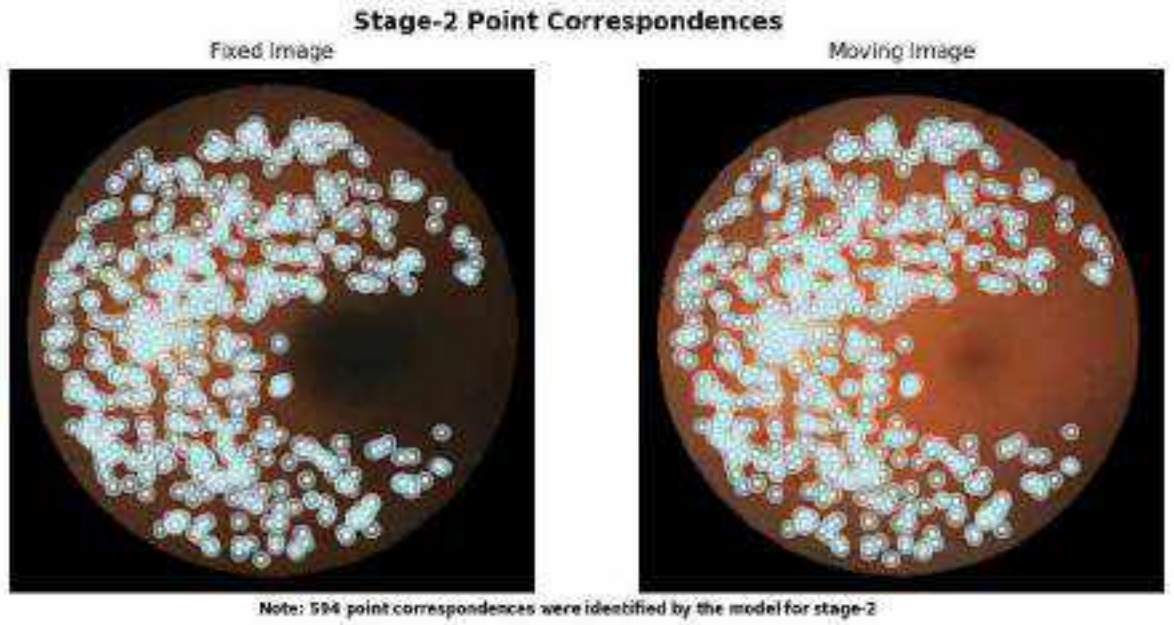

```
Mean Landmark Error for Case 32 Before Registration is 14.845049760325148 pixels
Mean Landmark Error for Case 32 After Registration is 1.7971856413469607 pixels
Case 33
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S34_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S34_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



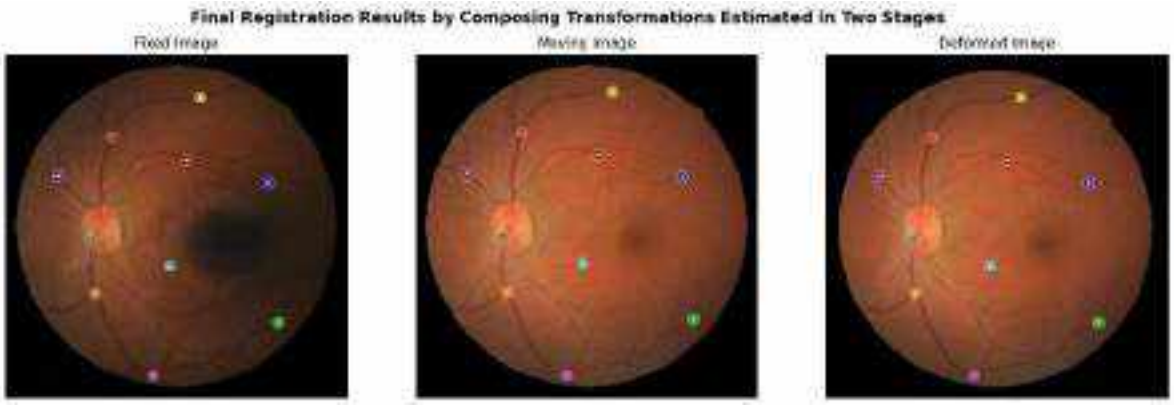

```
Homography Matrix:
[[ 9.80587395e-01 -7.47678309e-03  6.64825100e+00]
 [ 9.30974666e-03  9.82678020e-01  1.35409554e+01]
 [-2.79621360e-06  5.15599105e-06  1.00000000e+00]]
```



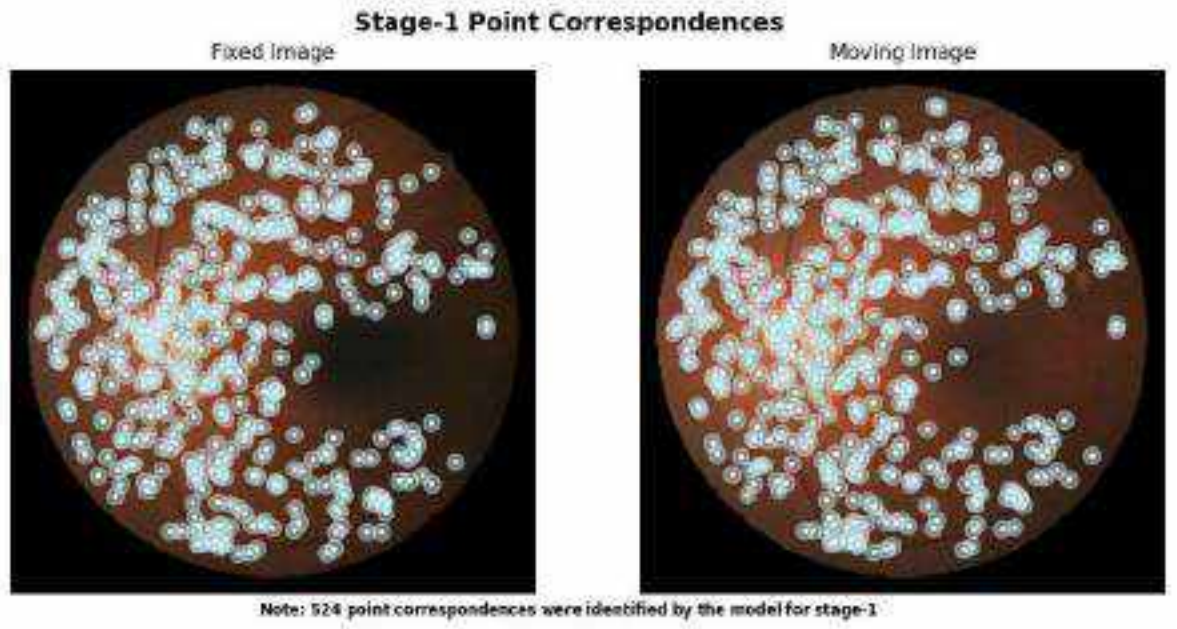

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



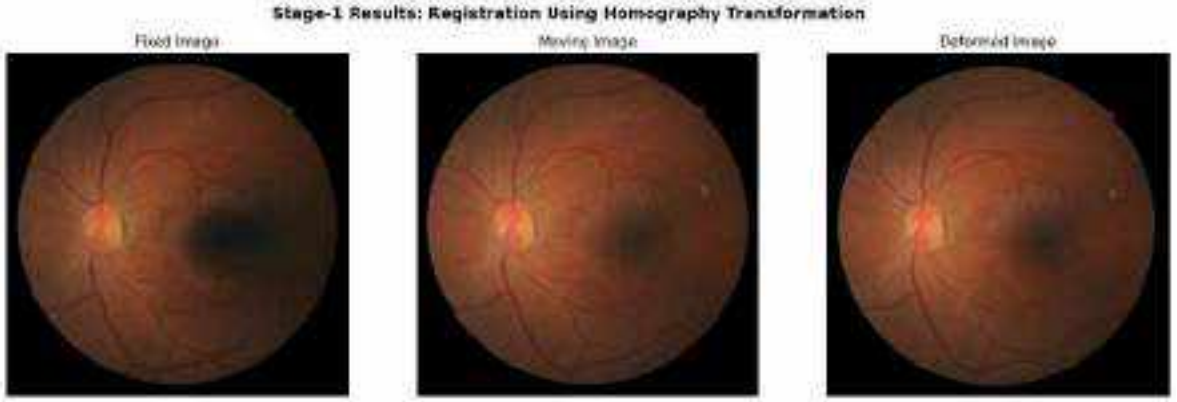
Note: 594 point correspondences were identified by the model for stage-2



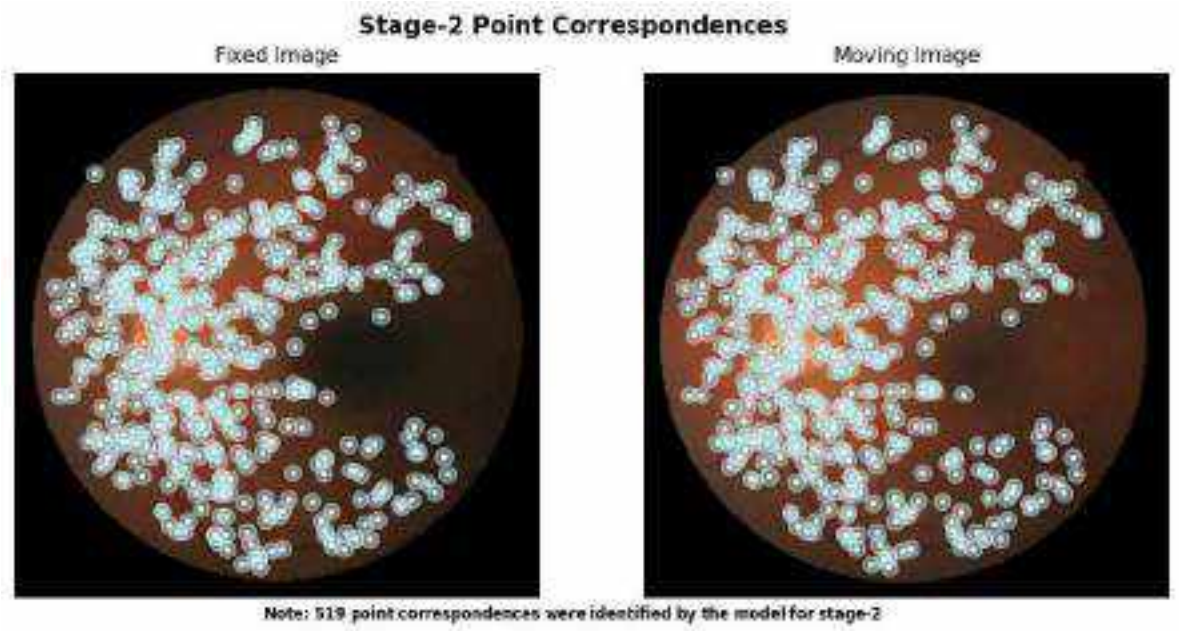
```
Mean Landmark Error for Case 33 Before Registration is 36.845584877272586 pixels
Mean Landmark Error for Case 33 After Registration is 2.035610137201705 pixels
Case 34
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S35_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S35_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



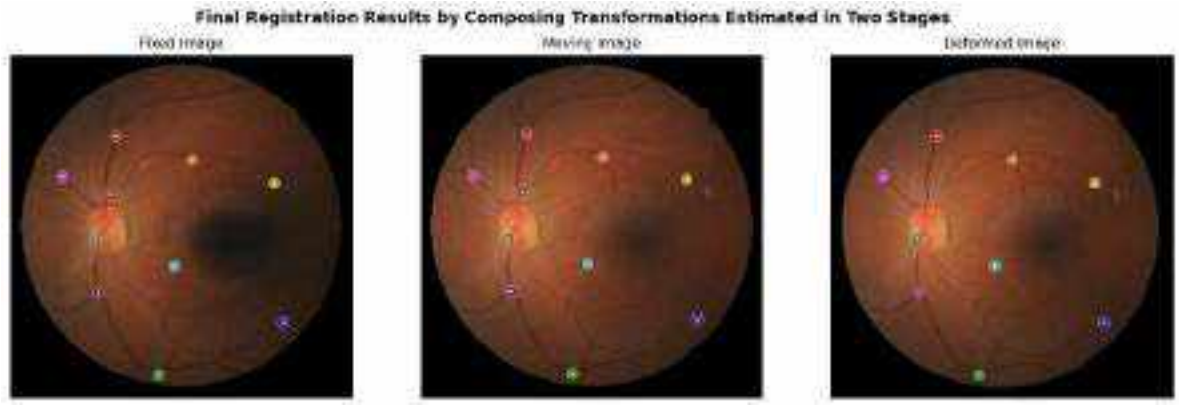

```
Homography Matrix:
[[ 9.91230744e-01 -1.35449889e-02  4.70708569e+00]
 [ 1.29265134e-02  9.94499526e-01  2.84155597e+00]
 [-1.75914041e-06  1.85142653e-06  1.00000000e+00]]
```



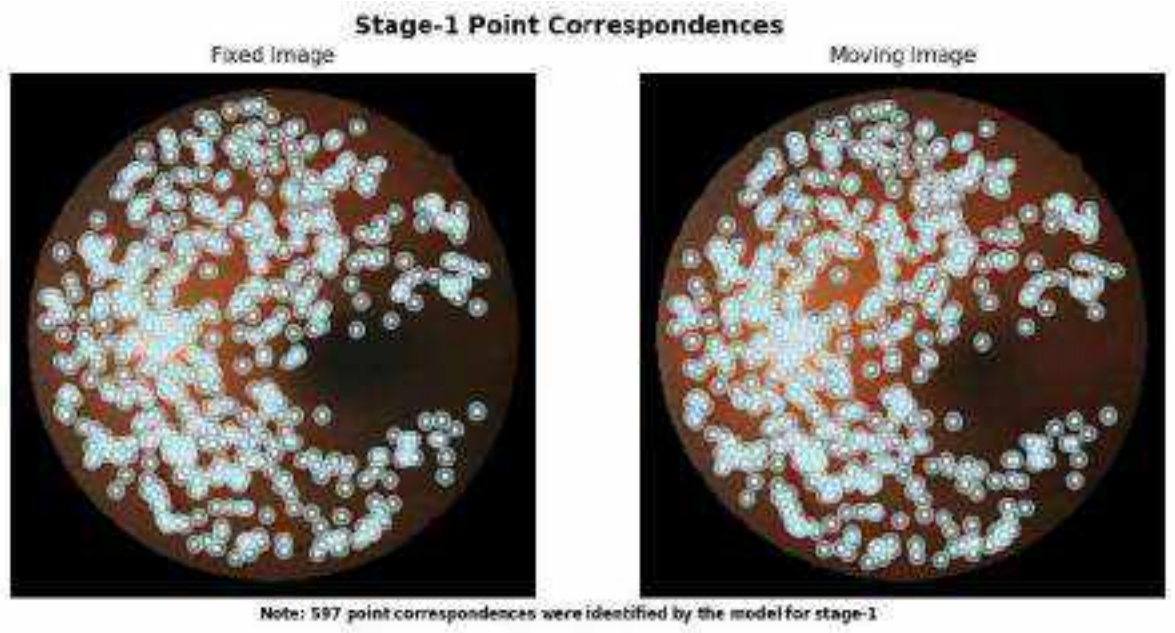

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 519 point correspondences were identified by the model for stage-2

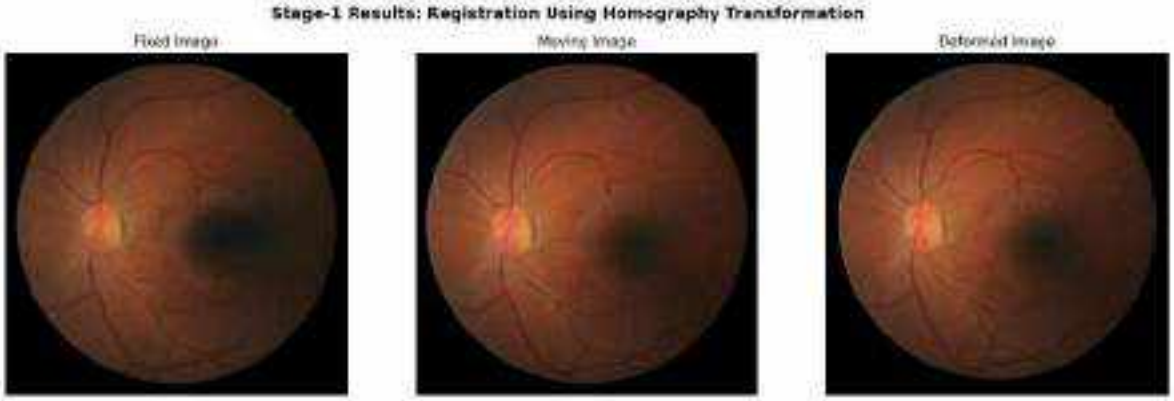



```
Mean Landmark Error for Case 34 Before Registration is 26.3937585286741 pixels
Mean Landmark Error for Case 34 After Registration is 1.4904961155607741 pixels
Case 35
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S36_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S36_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

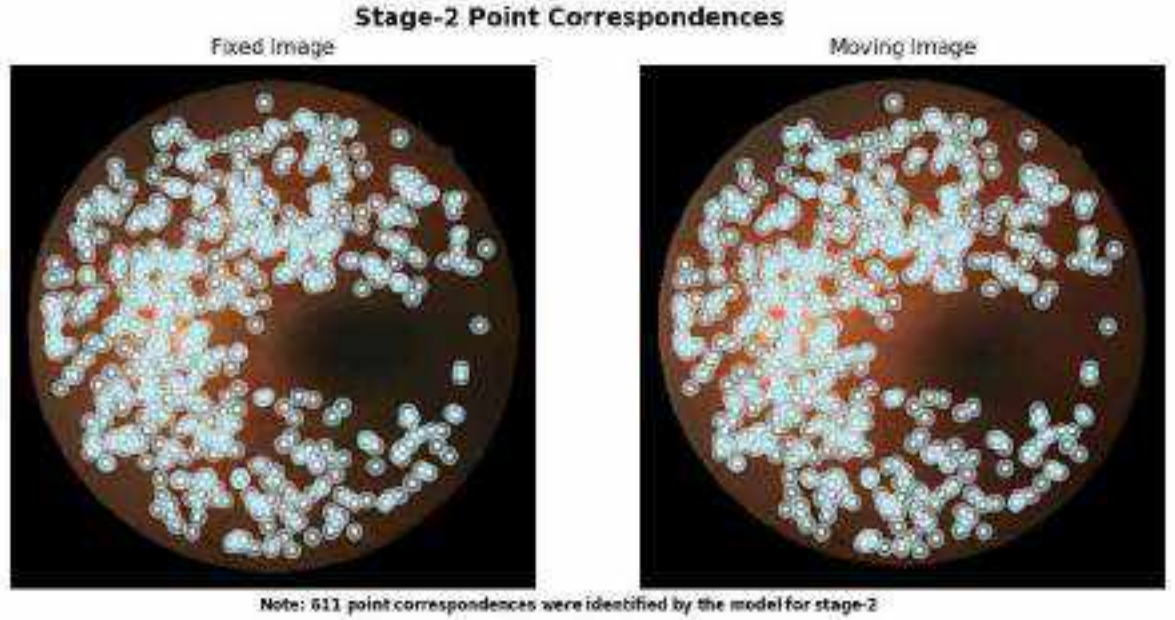


Note: 597 point correspondences were identified by the model for stage-1

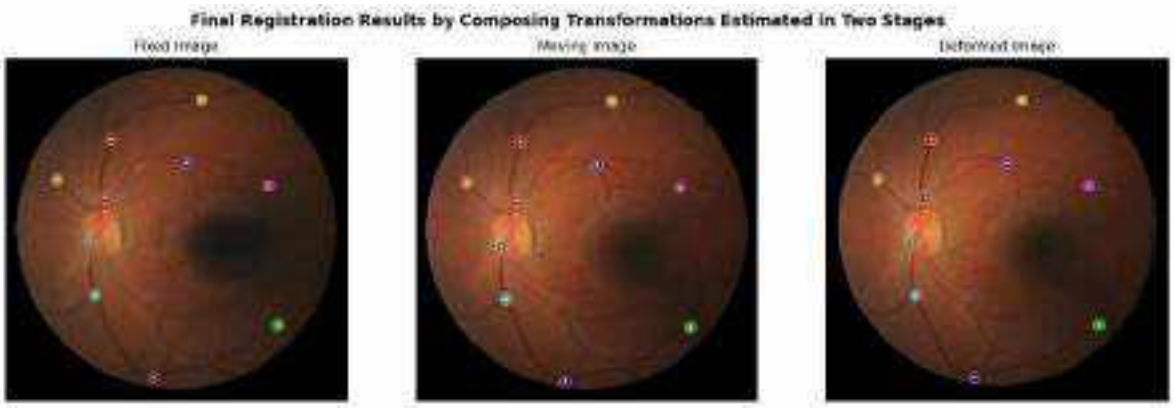
```
Homography Matrix:
[[ 9.92952715e-01 -6.43490886e-03  4.79469485e+00]
 [ 7.88299986e-03  9.90633306e-01 -6.13248777e+00]
 [-2.45602060e-06  1.44749344e-06  1.00000000e+00]]
```



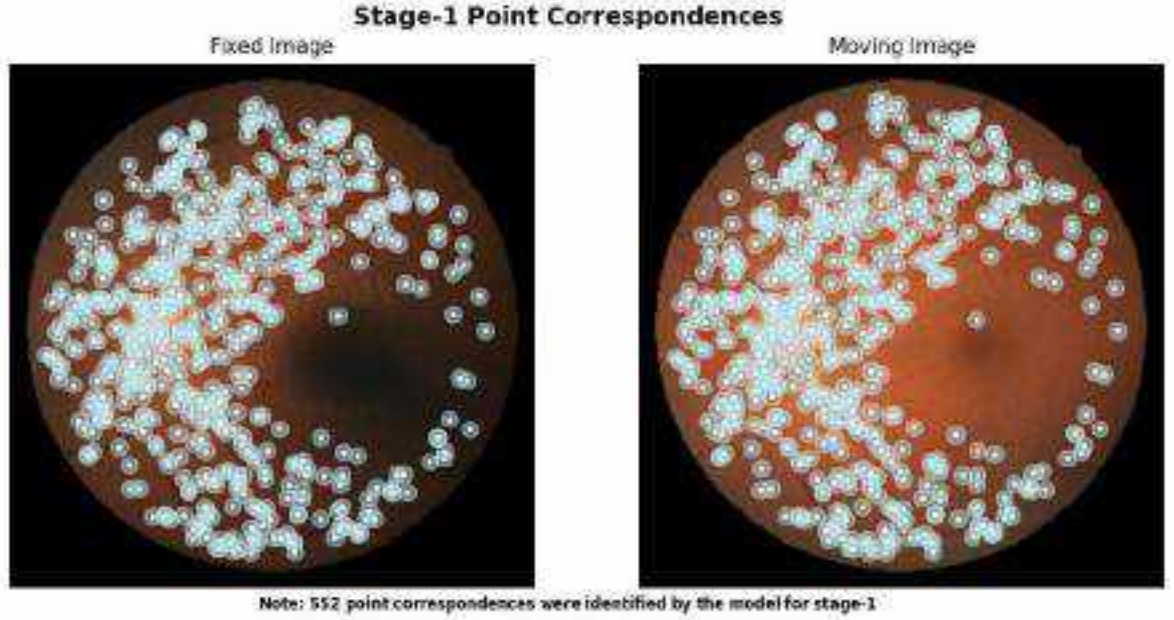

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



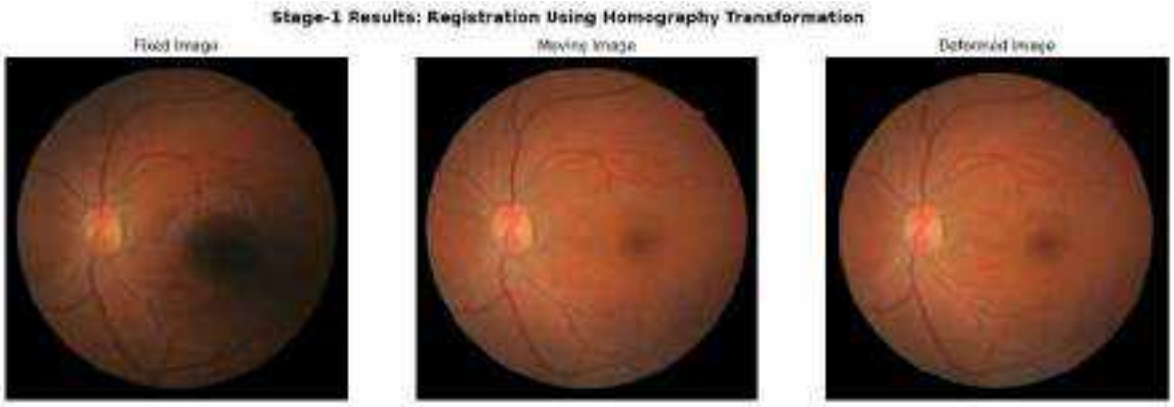
Note: 611 point correspondences were identified by the model for stage-2



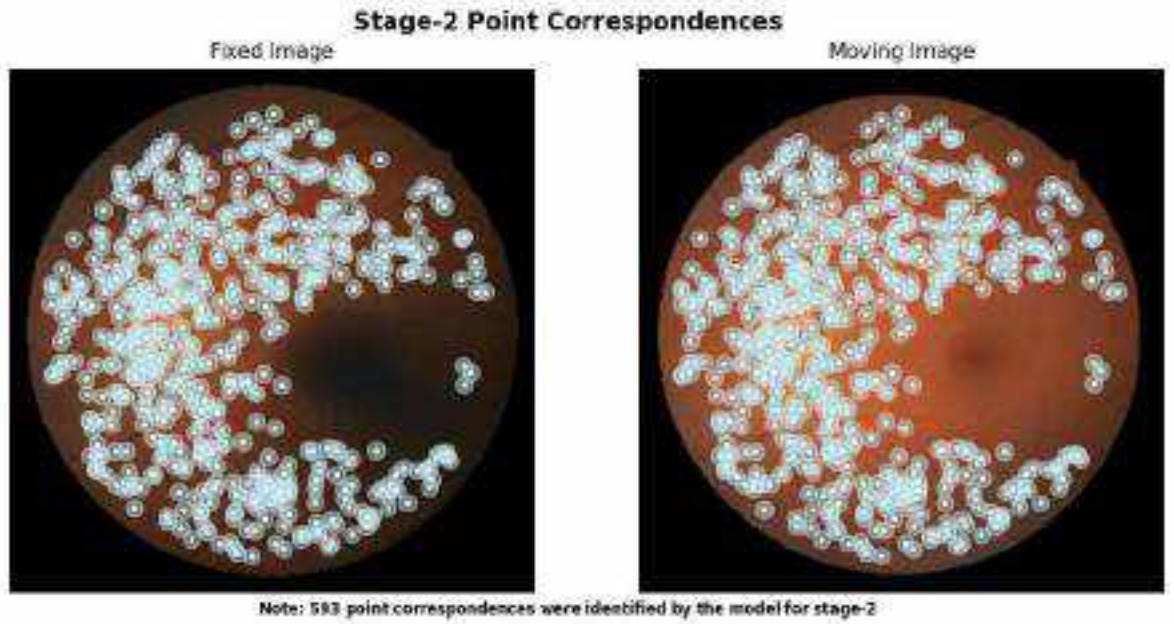

```
Mean Landmark Error for Case 35 Before Registration is 23.995287208513254 pixels
Mean Landmark Error for Case 35 After Registration is 2.0589663908801965 pixels
Case 36
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S37_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S37_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



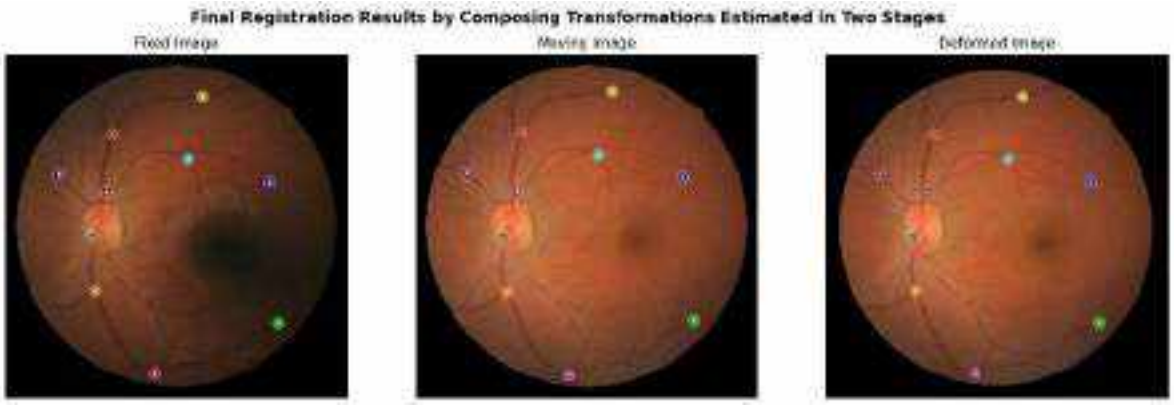

```
Homography Matrix:
[[ 9.77477840e-01 -2.06337523e-02  1.37451404e+01]
 [ 1.66849634e-02  9.71756476e-01  9.03868324e+00]
 [-1.57223603e-06 -7.11966323e-06  1.00000000e+00]]
```



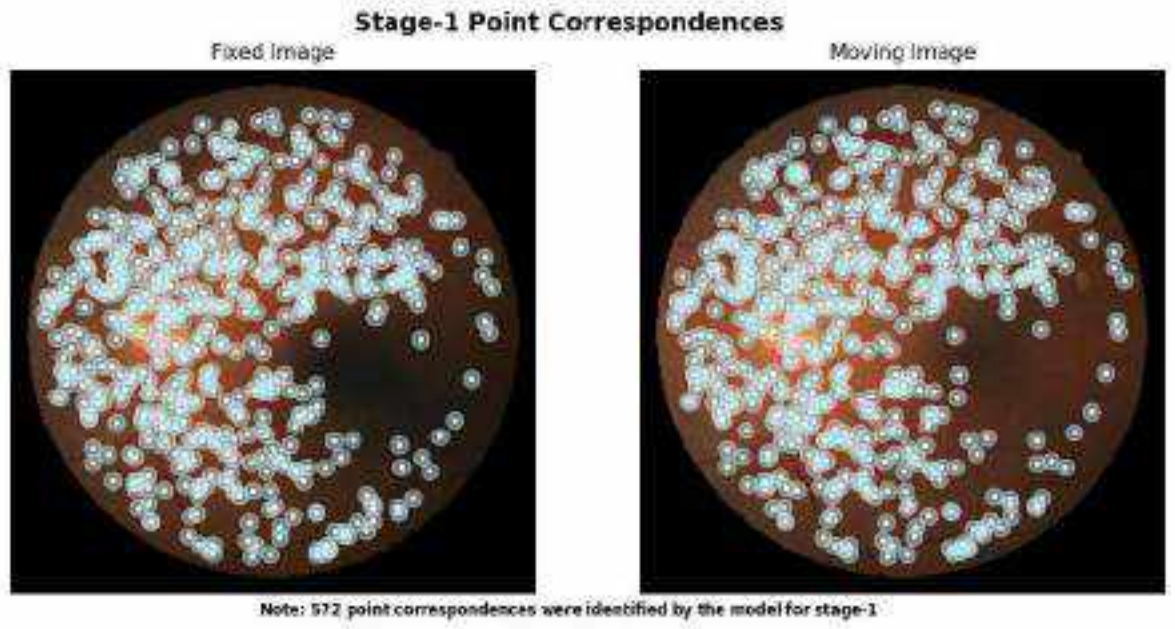

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



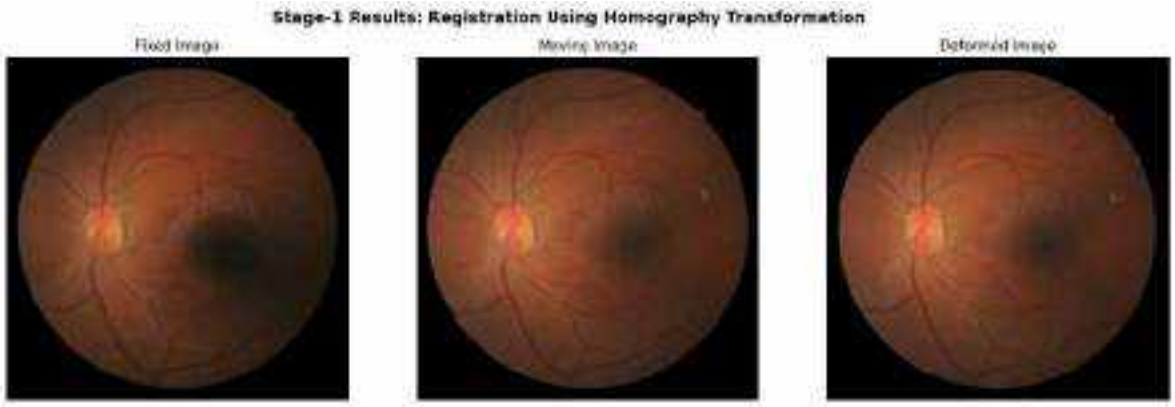
Note: 593 point correspondences were identified by the model for stage-2



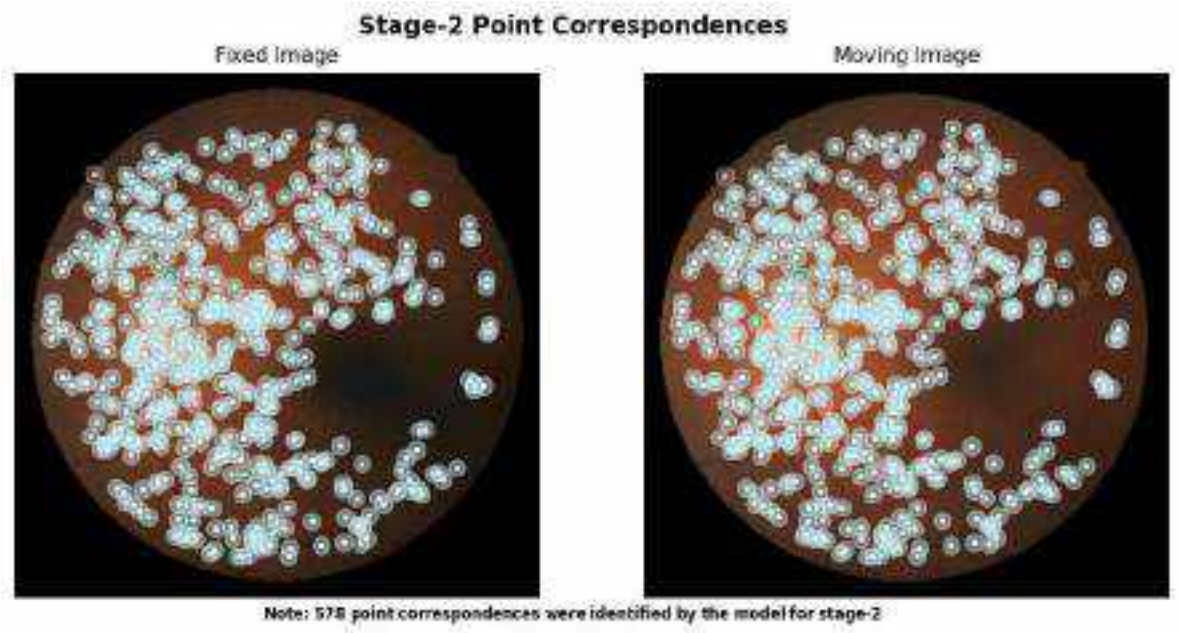
```
Mean Landmark Error for Case 36 Before Registration is 28.82230687890288 pixels
Mean Landmark Error for Case 36 After Registration is 1.9998826962047929 pixels
Case 37
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S38_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S38_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



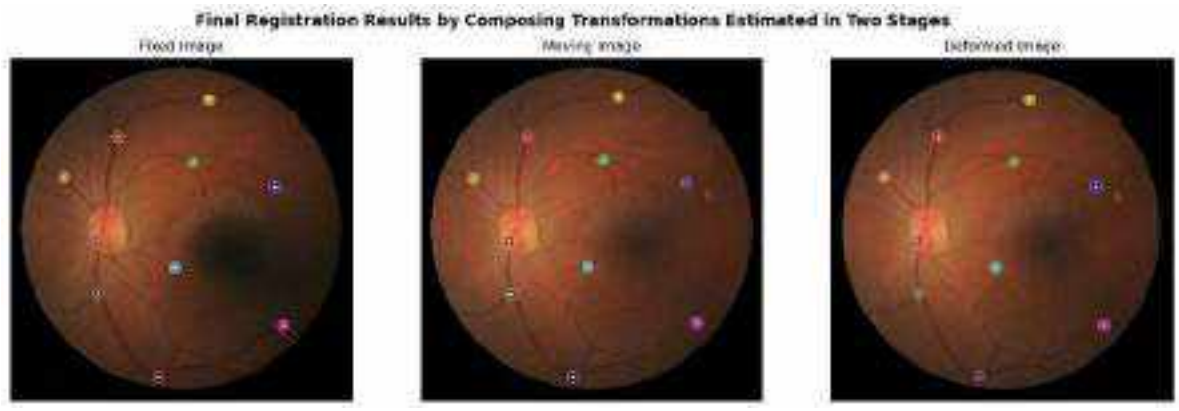

```
Homography Matrix:
[[ 9.92797241e-01 -2.18665820e-02  9.61948472e+00]
 [ 2.13509995e-02  9.89848608e-01 -2.64879829e+00]
 [ 1.16924436e-06 -3.49681894e-06  1.00000000e+00]]
```



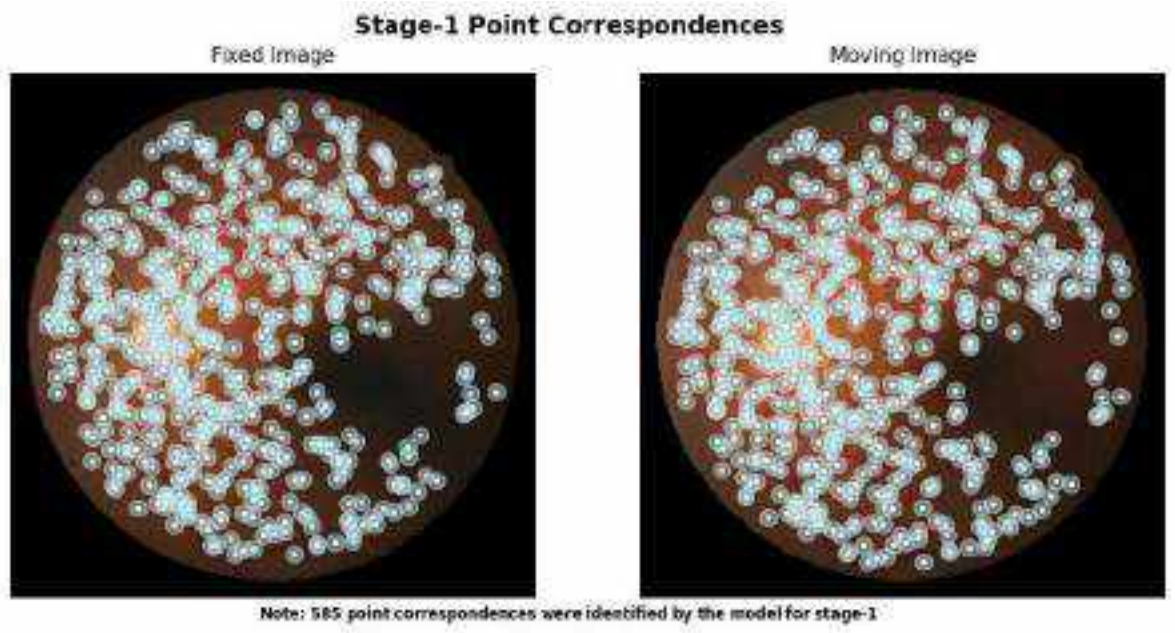

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 578 point correspondences were identified by the model for stage-2

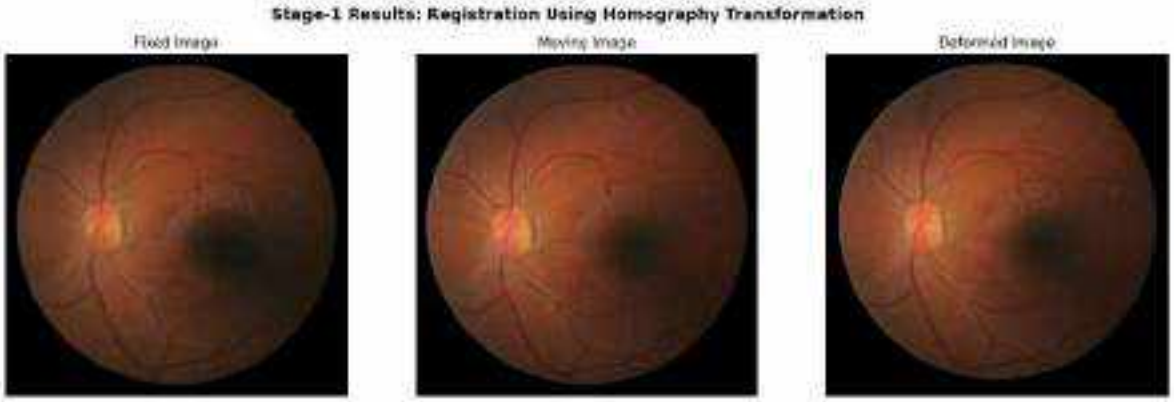



```
Mean Landmark Error for Case 37 Before Registration is 23.17161960561858 pixels
Mean Landmark Error for Case 37 After Registration is 1.8842939040862395 pixels
Case 38
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S39_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S39_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

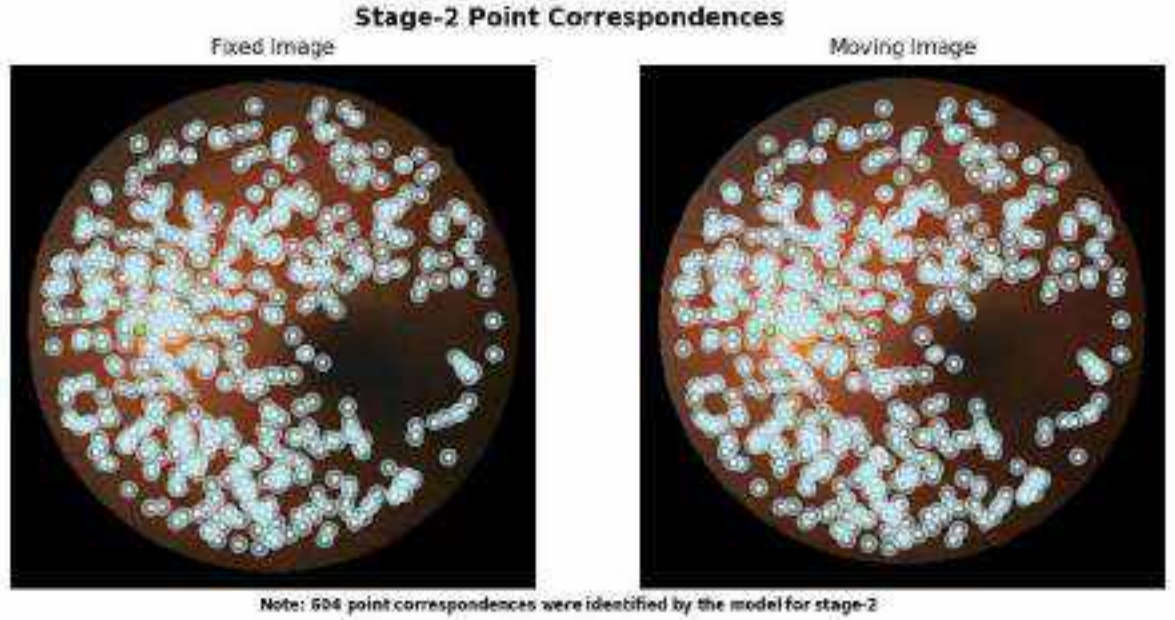


Note: 585 point correspondences were identified by the model for stage-1

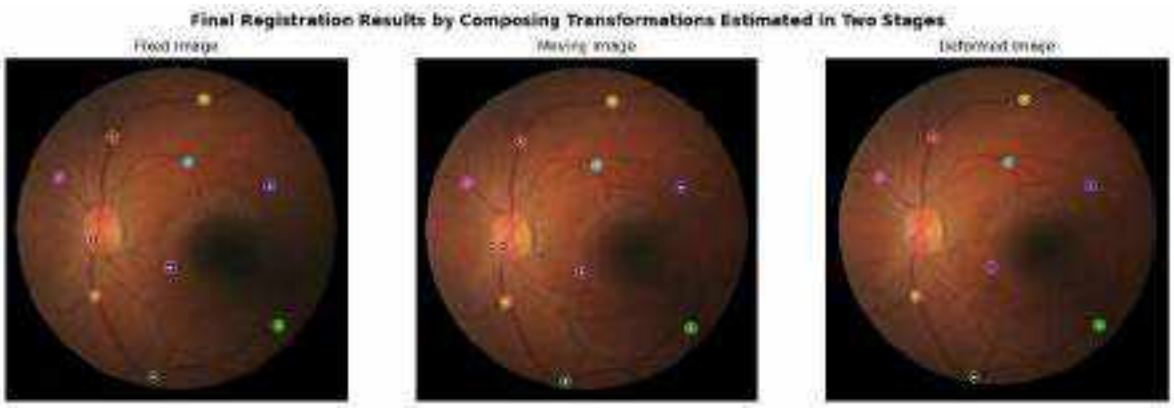
```
Homography Matrix:
[[ 9.91429416e-01 -1.84709694e-02  1.22654314e+01]
 [ 1.93068755e-02  9.87472451e-01 -1.37595916e+01]
 [-5.72886546e-08 -3.79322275e-06  1.00000000e+00]]
```



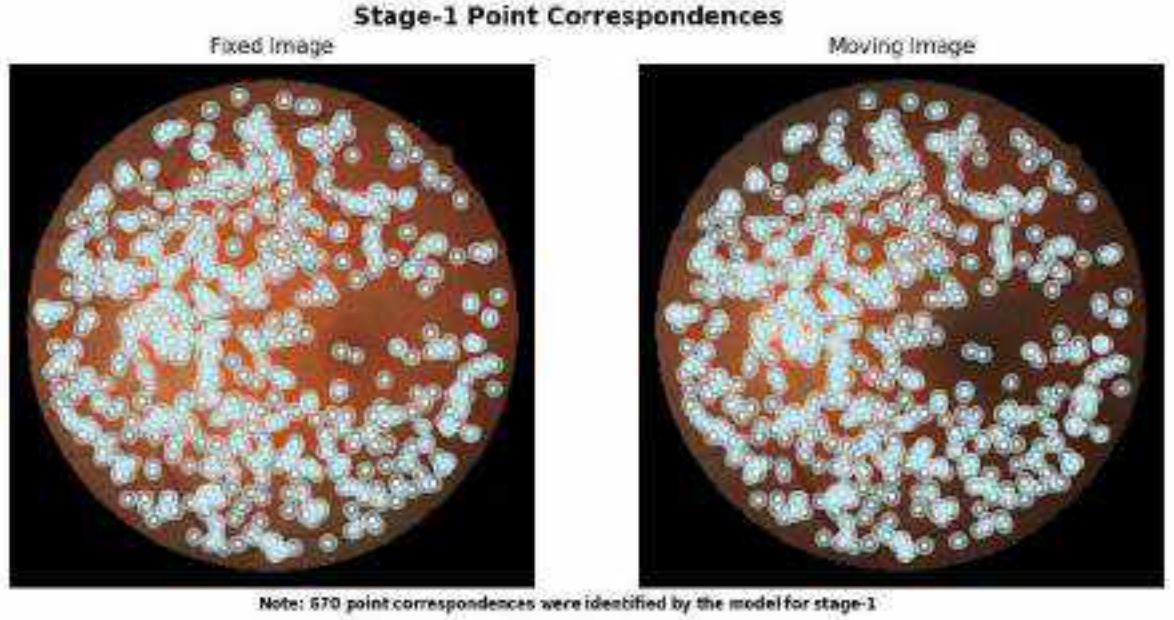

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



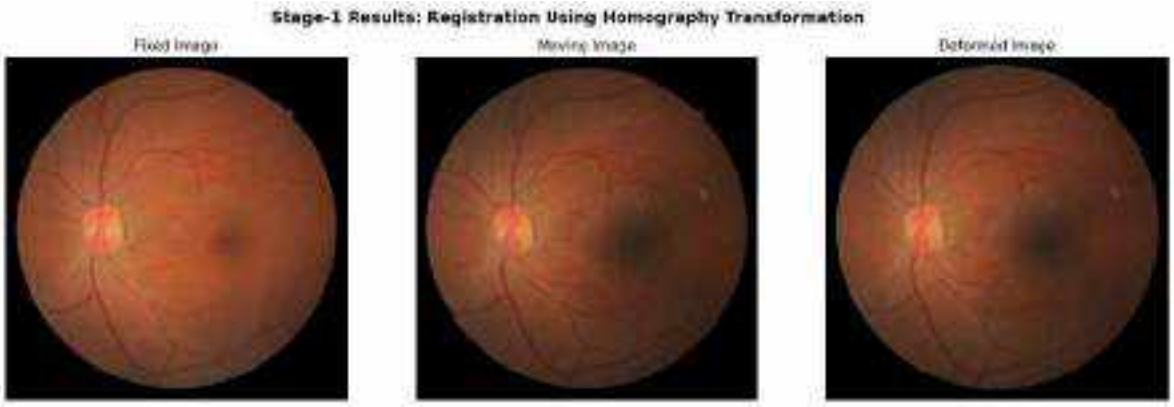
Note: 604 point correspondences were identified by the model for stage-2



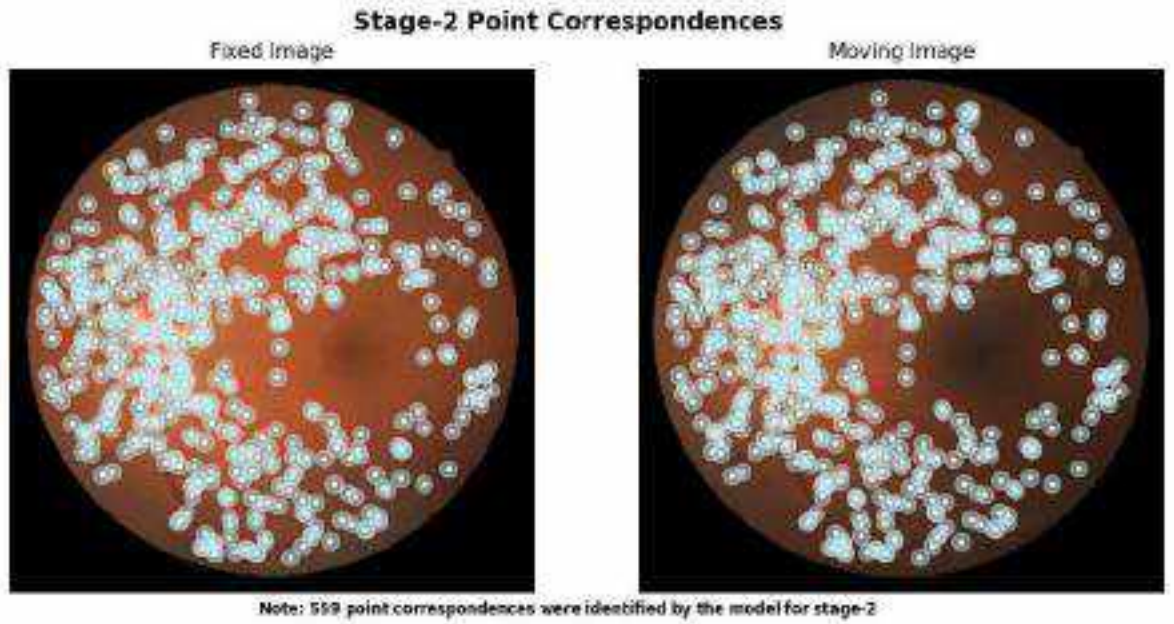

```
Mean Landmark Error for Case 38 Before Registration is 37.197752278014626 pixels
Mean Landmark Error for Case 38 After Registration is 1.6575415470257873 pixels
Case 39
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S40_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S40_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



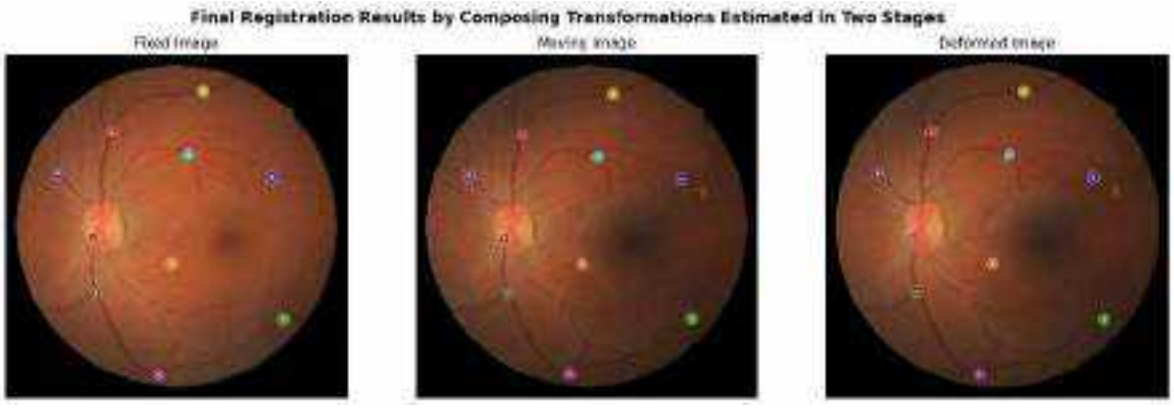

```
Homography Matrix:
[[ 1.00981360e+00 -1.46824193e-03 -3.34956585e+00]
 [ 2.33288653e-03  1.01394680e+00 -1.00766173e+01]
 [-7.80540609e-07  1.62987429e-06  1.00000000e+00]]
```



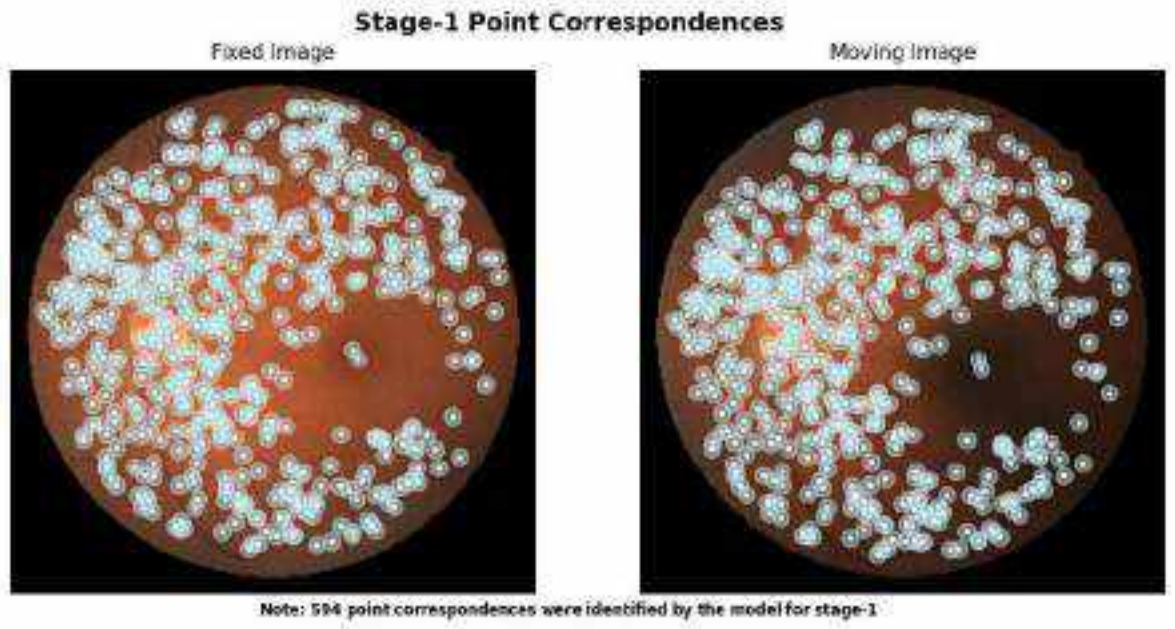

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



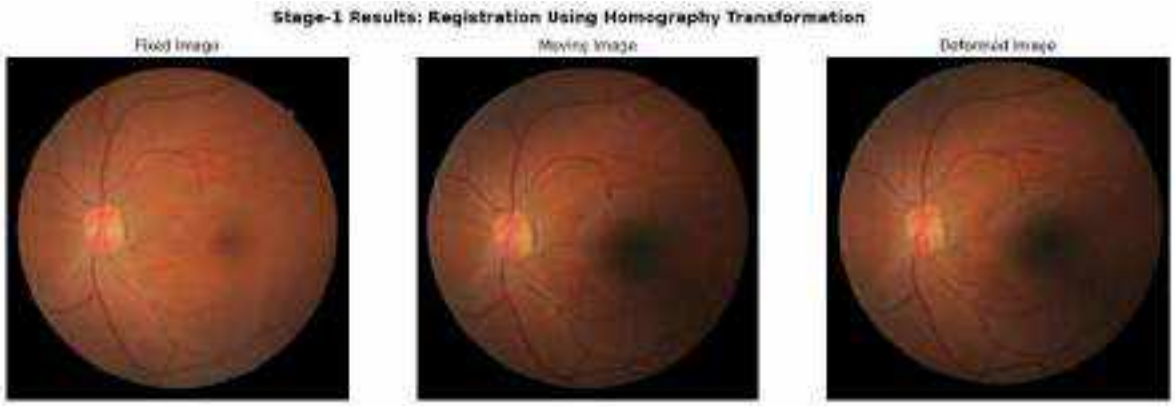
Note: 559 point correspondences were identified by the model for stage-2



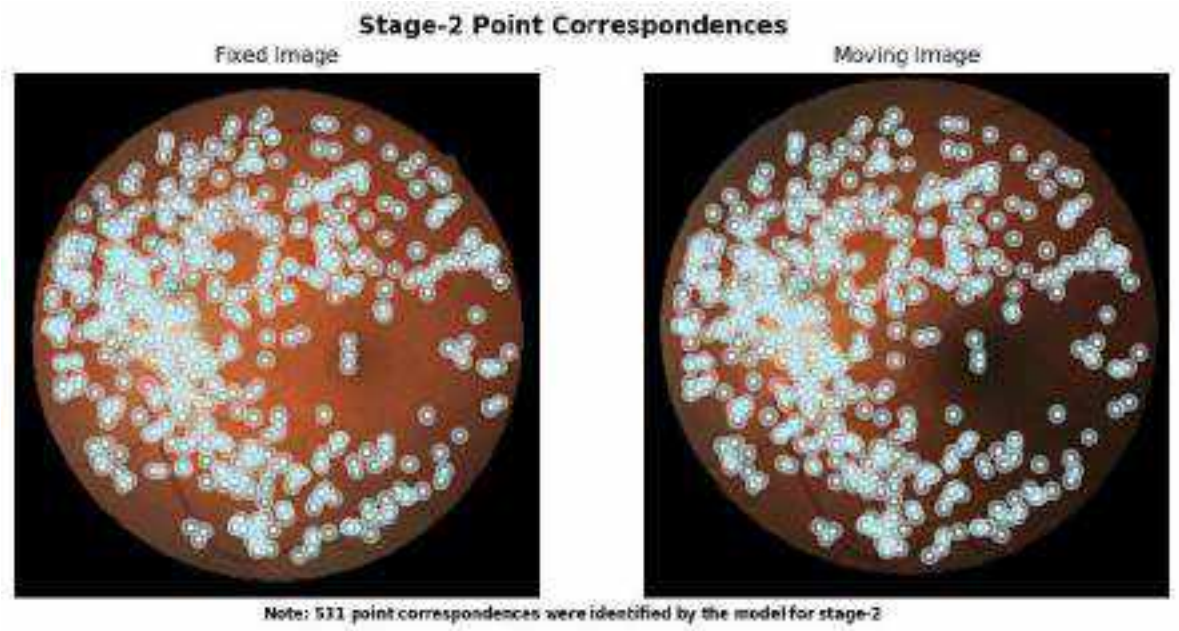
```
Mean Landmark Error for Case 39 Before Registration is 15.212162992606906 pixels
Mean Landmark Error for Case 39 After Registration is 1.1396889859551709 pixels
Case 40
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S41_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S41_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

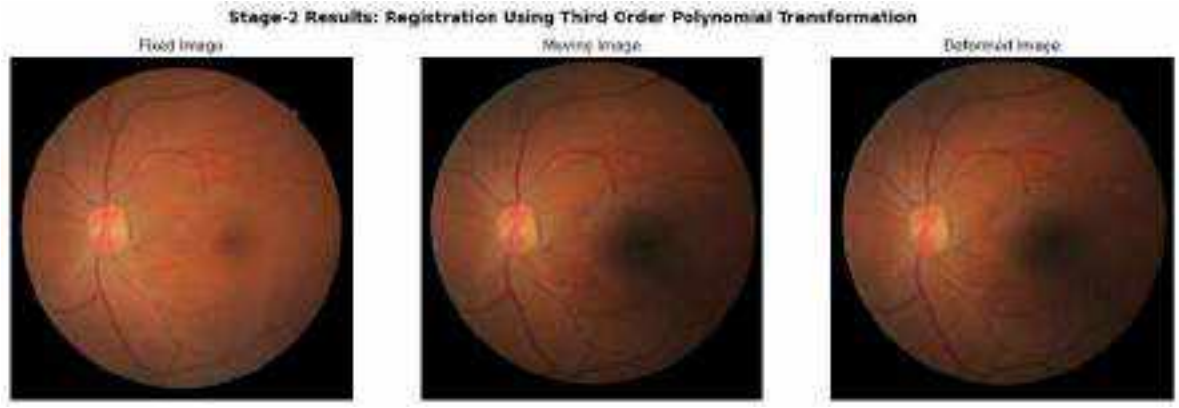


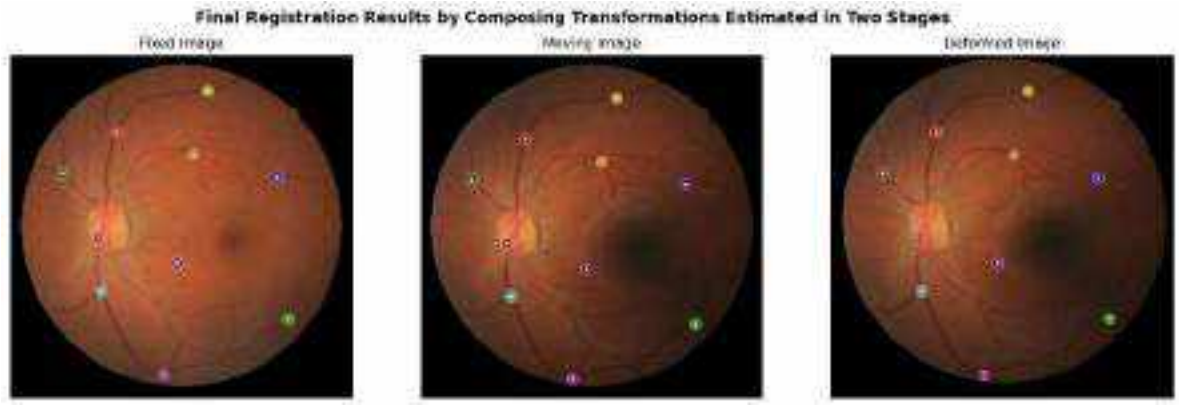

```
Homography Matrix:
[[ 1.00413471e+00 -1.90854782e-03  8.01843723e-02]
 [-3.72855977e-03  9.98894113e-01 -1.70215288e+01]
 [-6.55281211e-06 -7.65062709e-06  1.00000000e+00]]
```



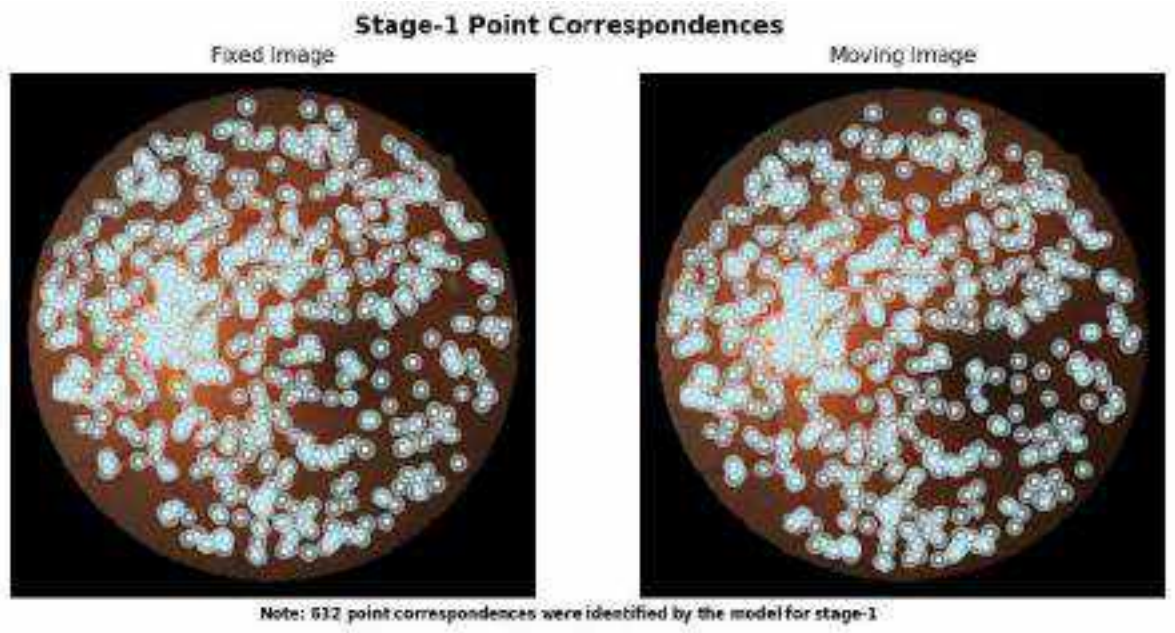

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```







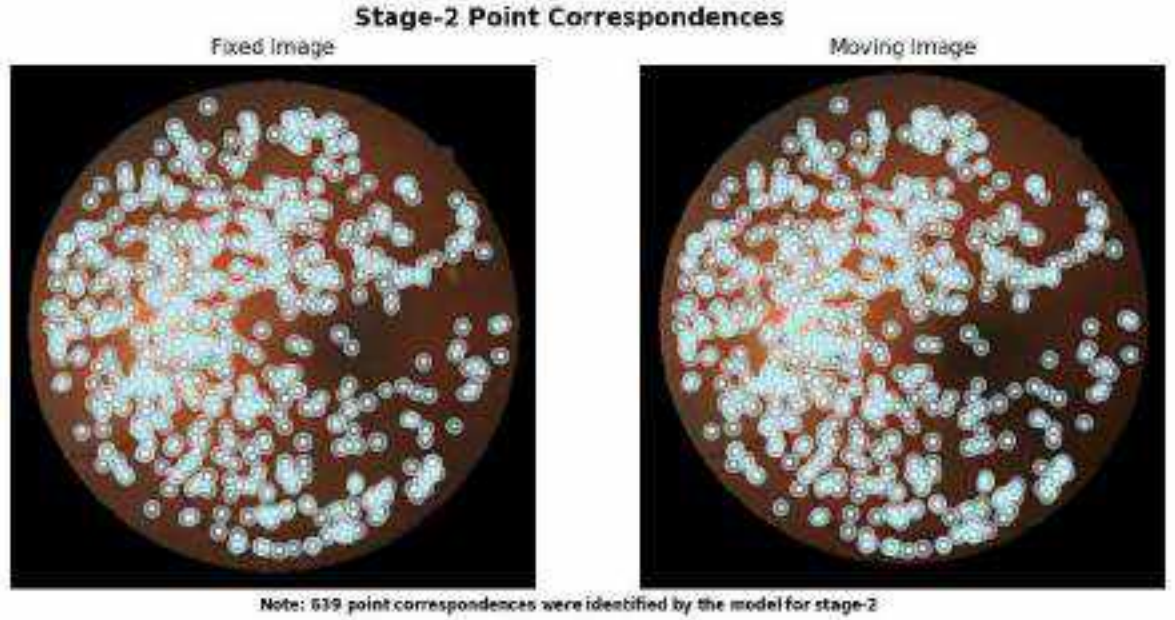

```
Mean Landmark Error for Case 40 Before Registration is 53.143720080523996 pixels
Mean Landmark Error for Case 40 After Registration is 2.4813274388876621 pixels
Case 41
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S42_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S42_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



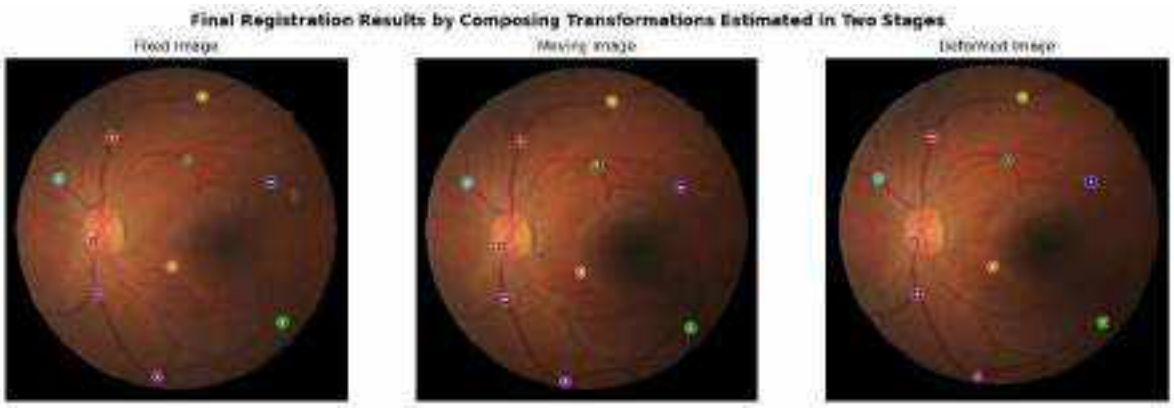
Note: 612 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.96130892e-01  1.28442404e-03  3.50459818e+00]
 [-3.74078337e-03  9.92557473e-01 -9.89848524e+00]
 [ 7.09872402e-07 -6.26763750e-06  1.00000000e+00]]
```

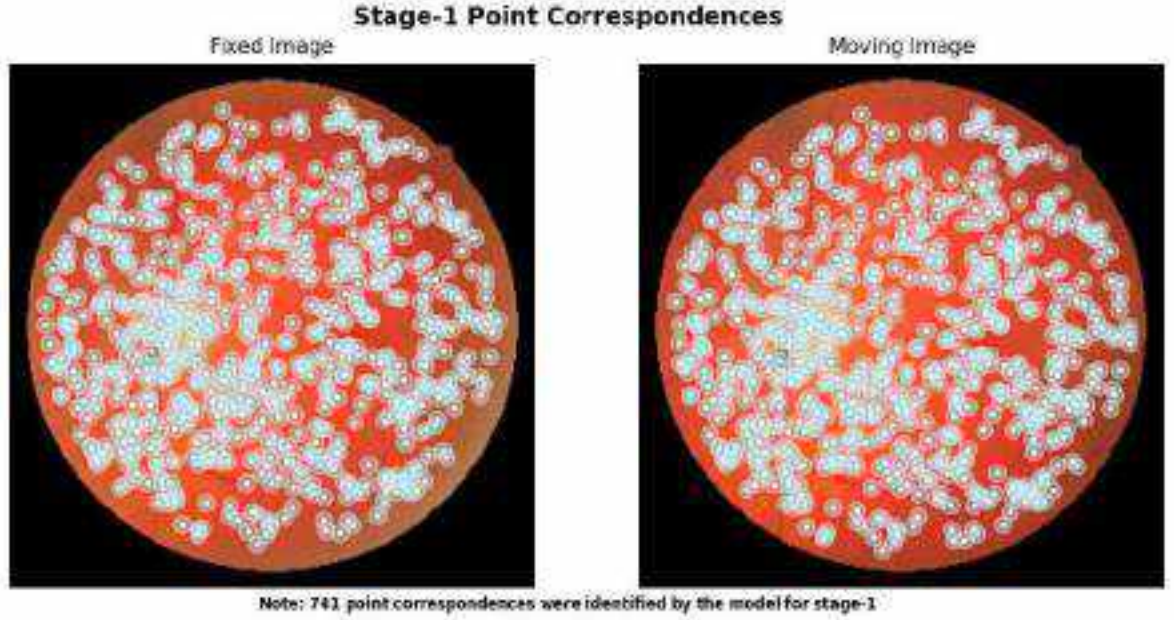

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 639 point correspondences were identified by the model for stage-2

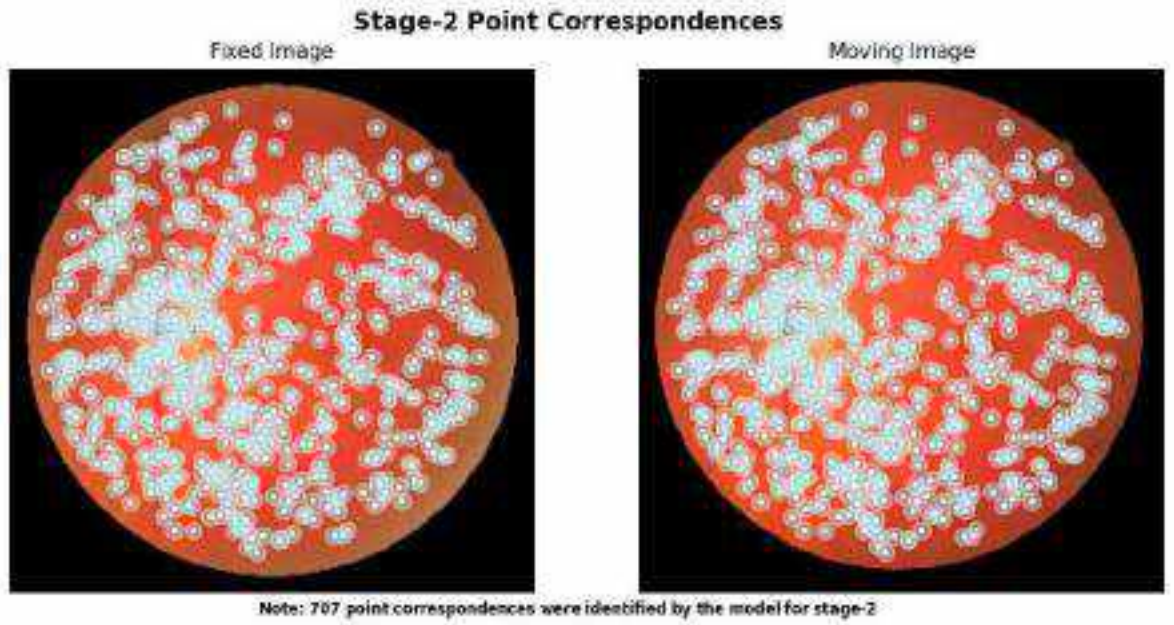



```
Mean Landmark Error for Case 41 Before Registration is 43.15196414392587 pixels
Mean Landmark Error for Case 41 After Registration is 1.8834352002313977 pixels
Case 42
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S43_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S43_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

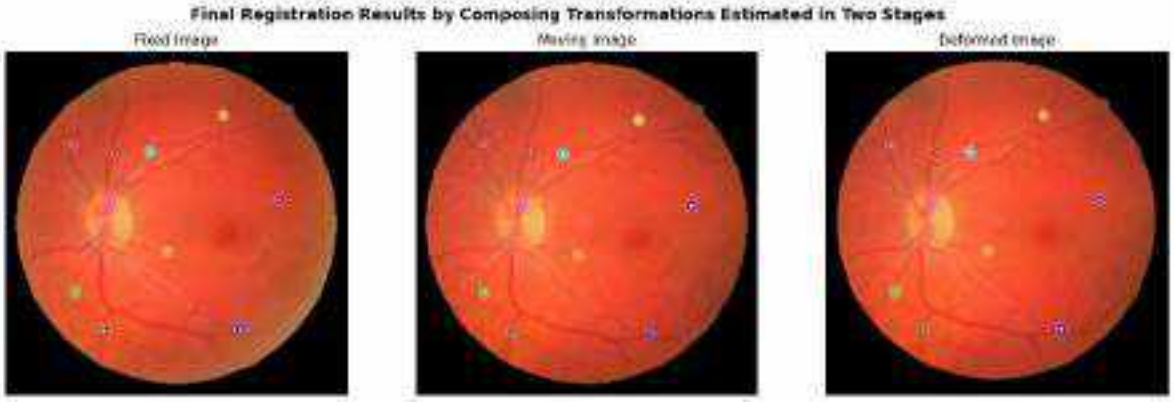



```
Homography Matrix:
[[ 9.85031979e-01  2.56545968e-02 -1.25257767e+01]
 [-3.45172405e-02  9.84417986e-01  1.06573841e+01]
 [-6.84368894e-06 -6.45982074e-06  1.00000000e+00]]
```

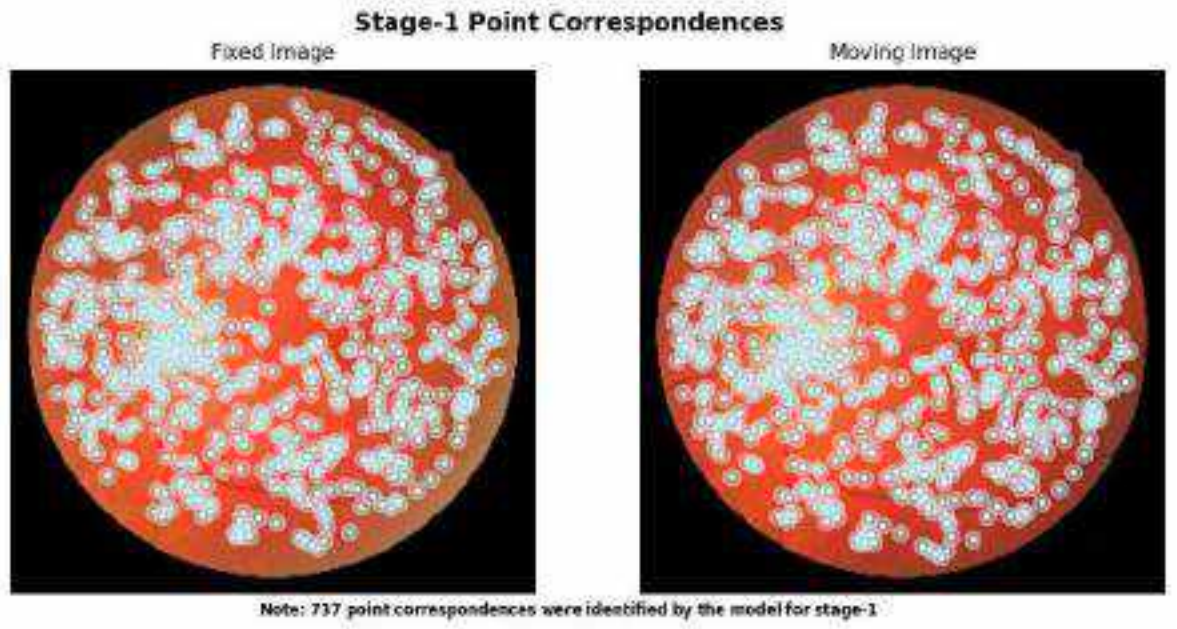

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



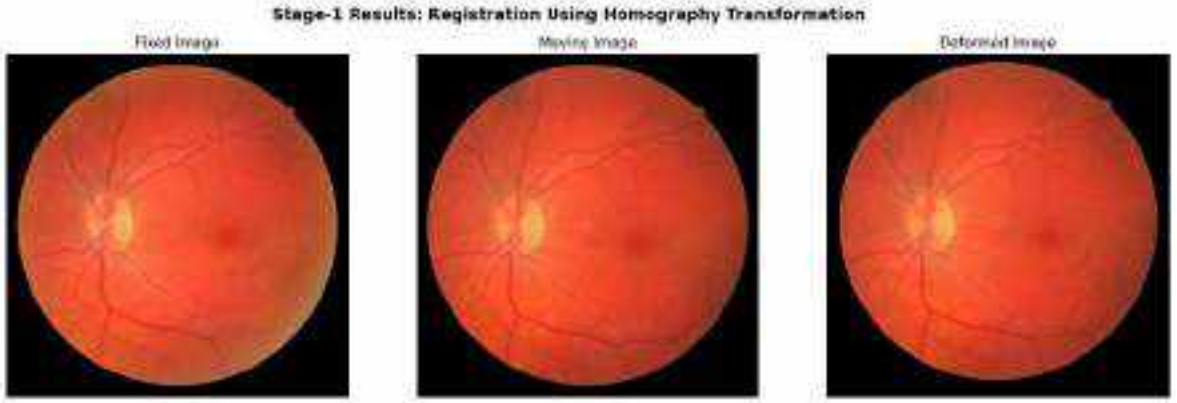
Note: 707 point correspondences were identified by the model for stage-2



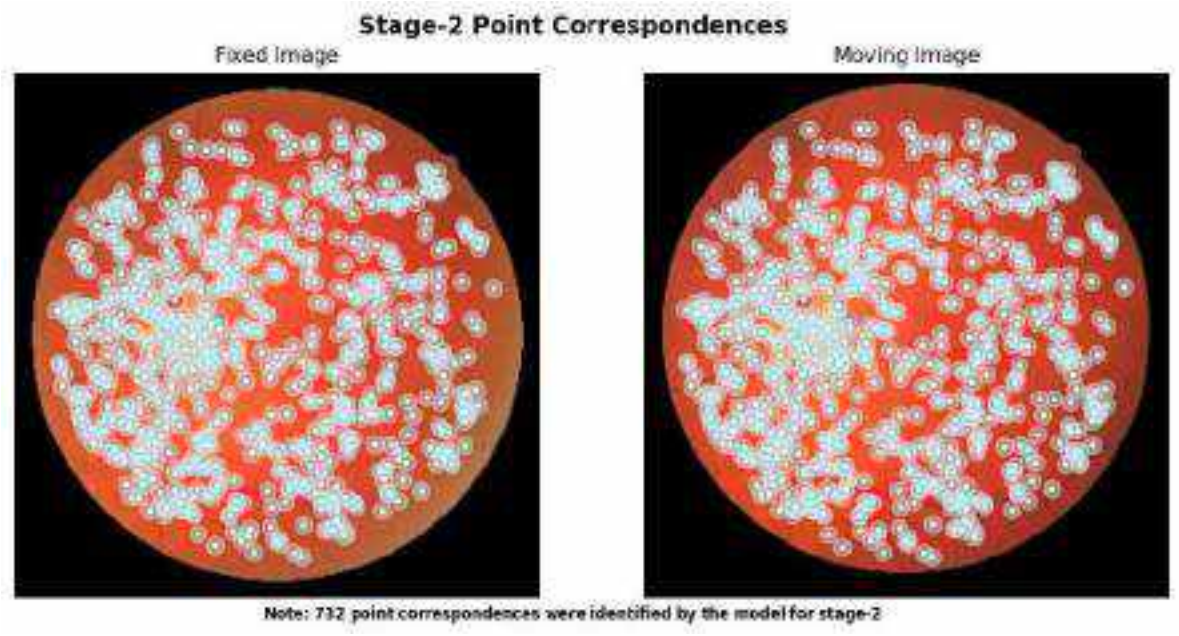
```
Mean Landmark Error for Case 42 Before Registration is 35.29090627584701 pixels
Mean Landmark Error for Case 42 After Registration is 1.8742141742624405 pixels
Case 43
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S44_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S44_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



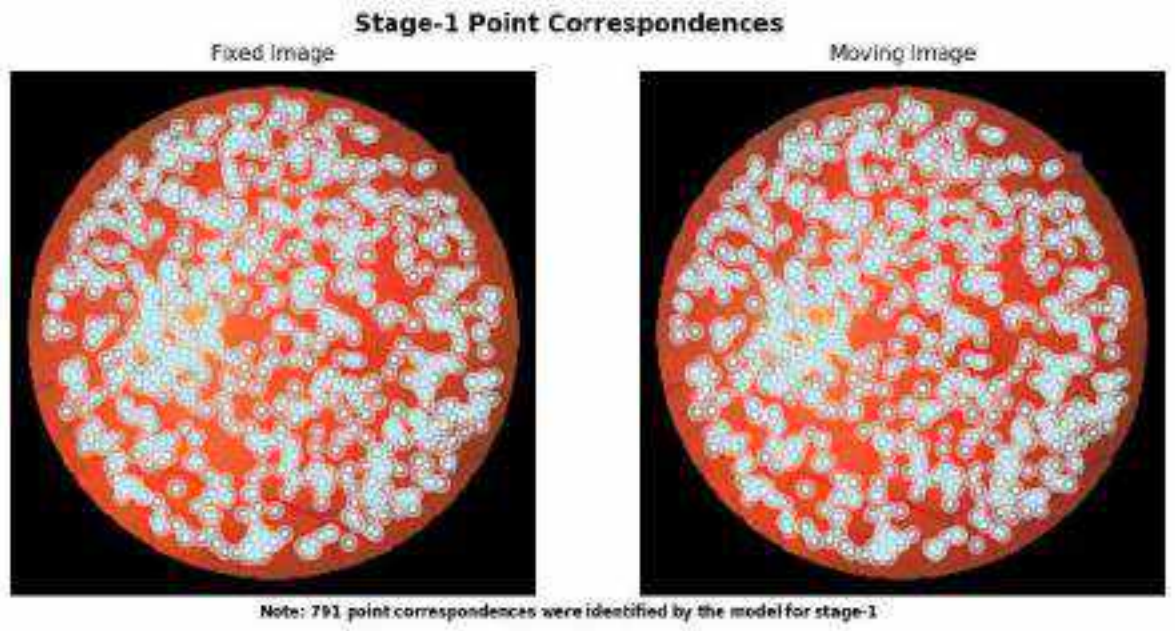

```
Homography Matrix:
[[ 9.92614871e-01  2.15116109e-02 -7.42432486e+00]
 [-2.83688842e-02  9.88488654e-01  5.19863261e+00]
 [ 1.23705444e-06 -5.83255709e-06  1.00000000e+00]]
```



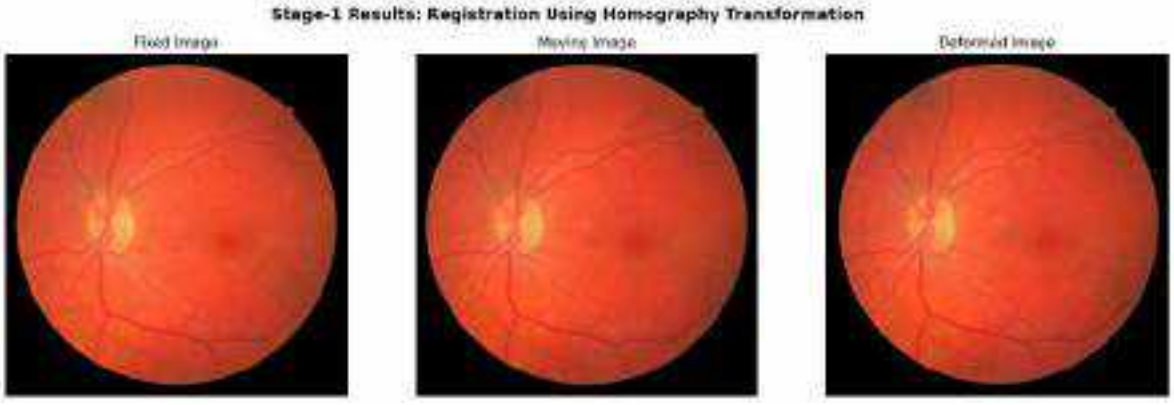

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



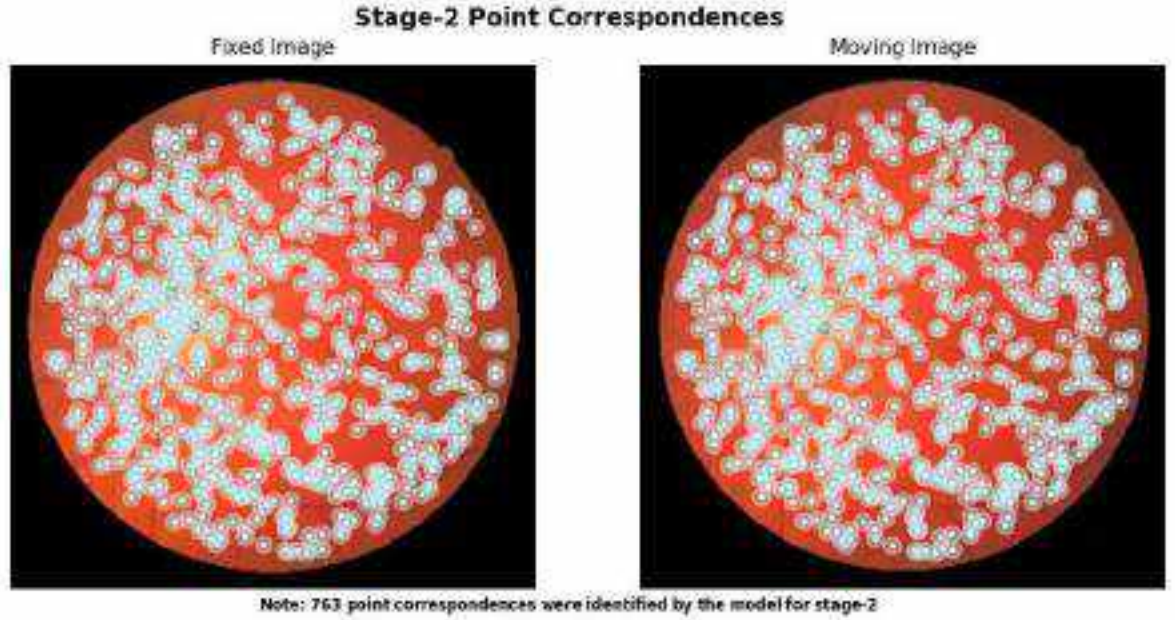

```
Mean Landmark Error for Case 43 Before Registration is 39.89714530340713 pixels
Mean Landmark Error for Case 43 After Registration is 1.6828077528136371 pixels
Case 44
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S45_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S45_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

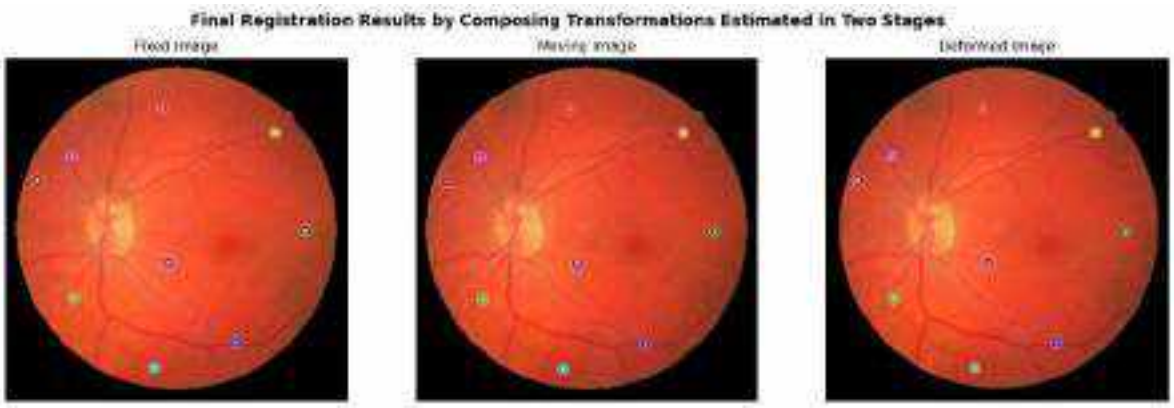



```
Homography Matrix:
[[ 9.96902949e-01 -5.06544757e-03  6.59950613e+00]
 [ 3.29310376e-03  9.95854856e-01 -2.11205167e+00]
 [ 1.88781061e-06 -3.93728018e-06  1.00000000e+00]]
```

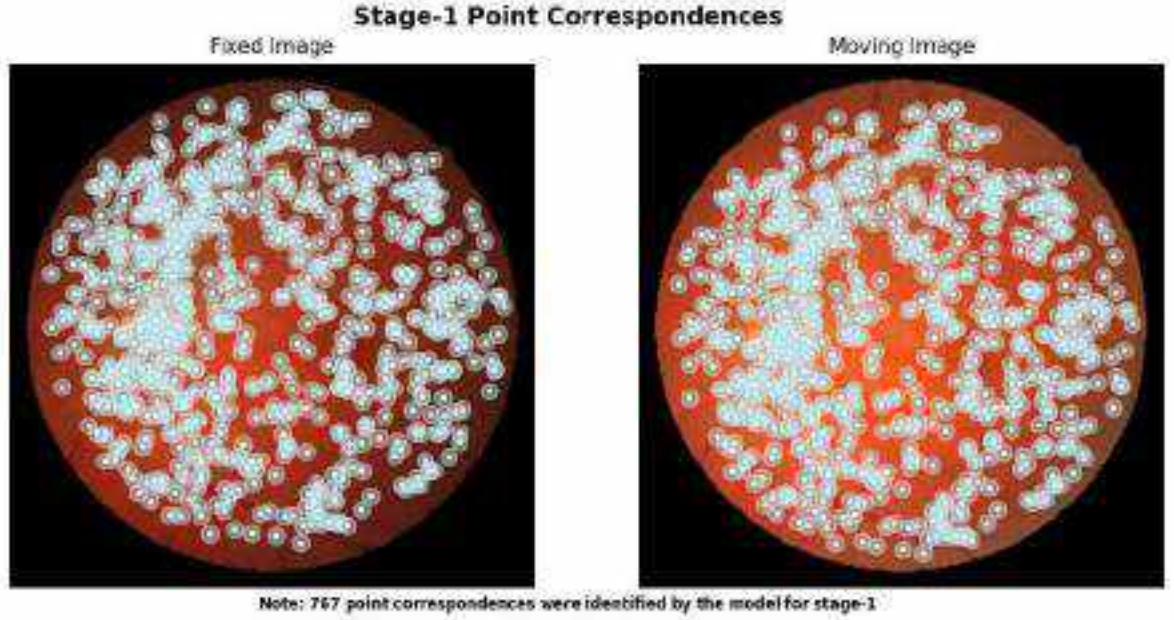



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

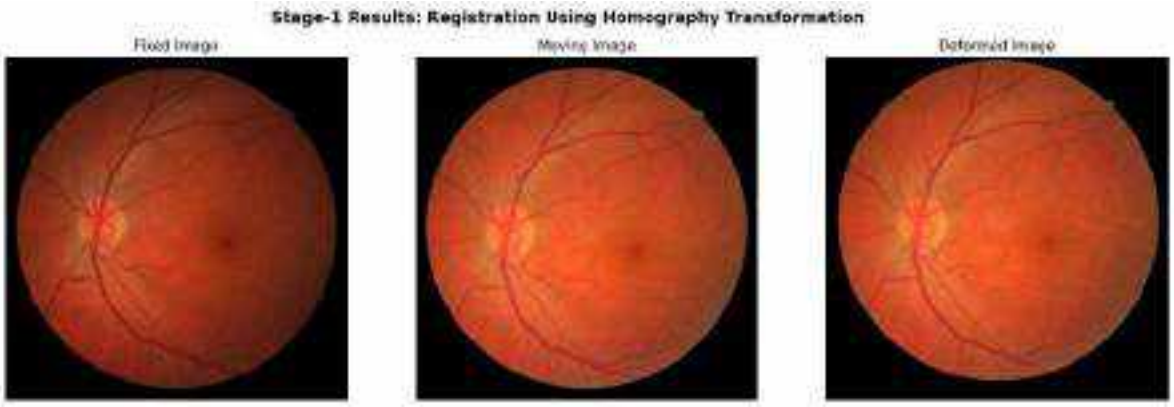




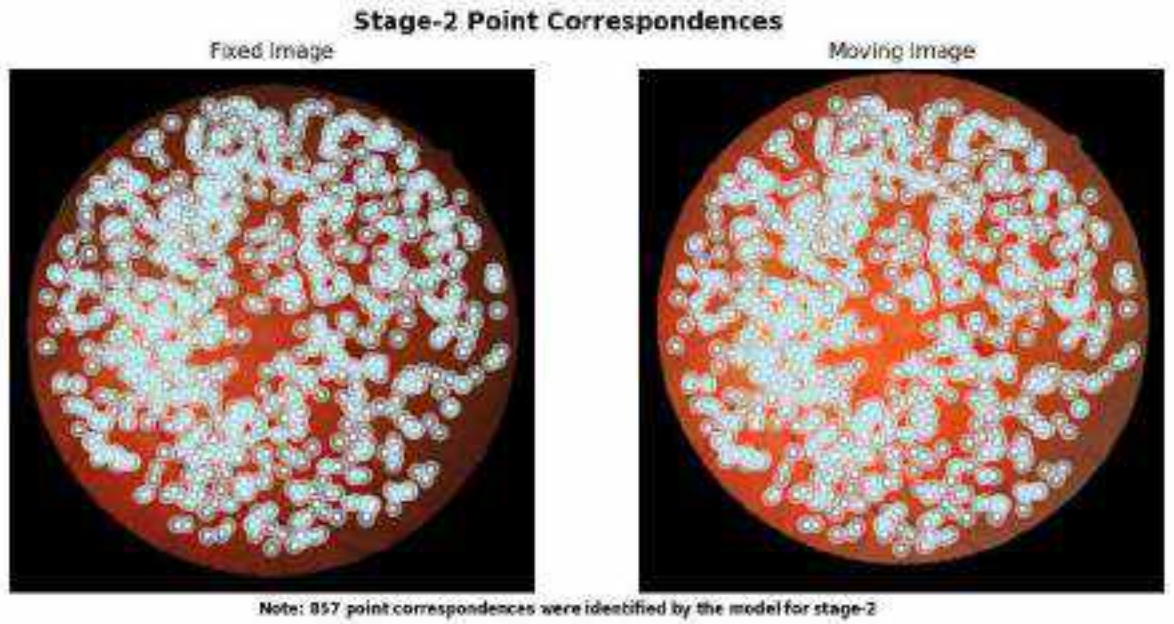

```
Mean Landmark Error for Case 44 Before Registration is 17.497467131309996 pixels
Mean Landmark Error for Case 44 After Registration is 1.5941208805219584 pixels
Case 45
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S46_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S46_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



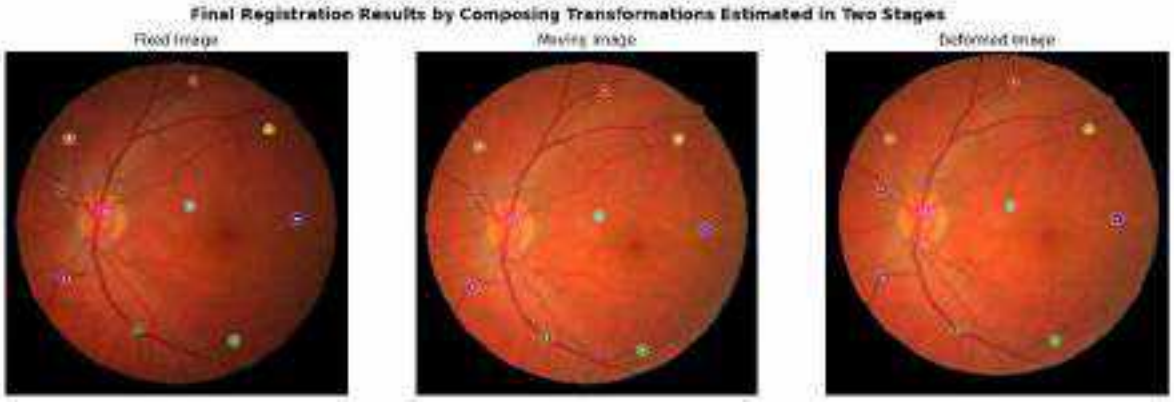

```
Homography Matrix:
[[ 9.92243160e-01 -3.02635488e-04  1.72486196e+00]
 [-1.20191227e-02  9.88793951e-01 -1.55985007e+01]
 [-6.15881614e-07 -2.22410625e-05  1.00000000e+00]]
```



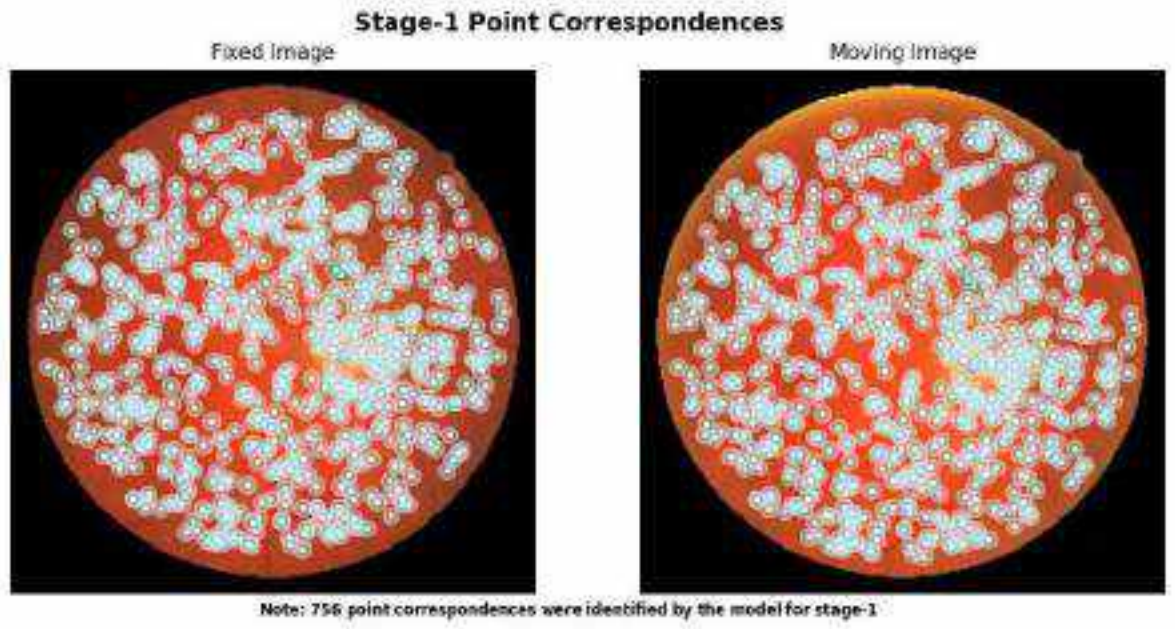

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 857 point correspondences were identified by the model for stage-2

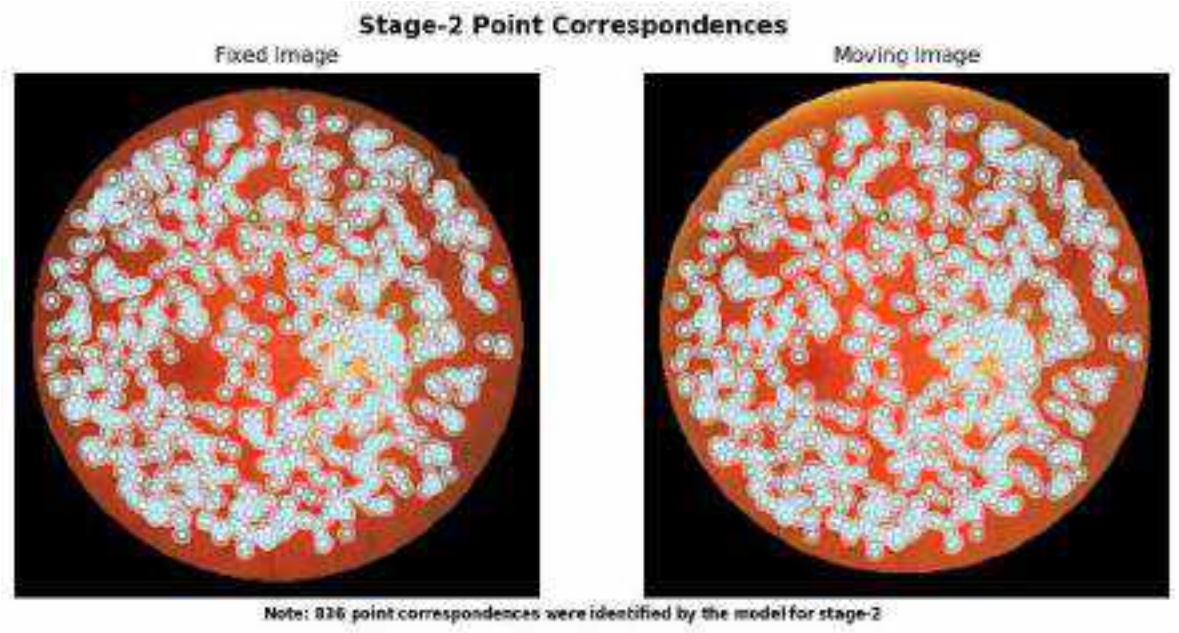


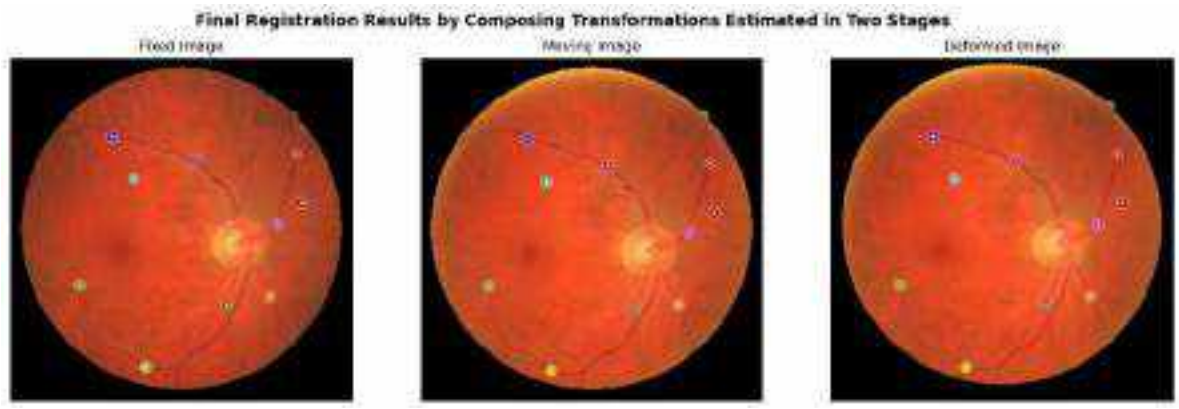
```
Mean Landmark Error for Case 45 Before Registration is 77.132447933322 pixels
Mean Landmark Error for Case 45 After Registration is 1.9429376054682543 pixels
Case 46
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S47_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S47_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



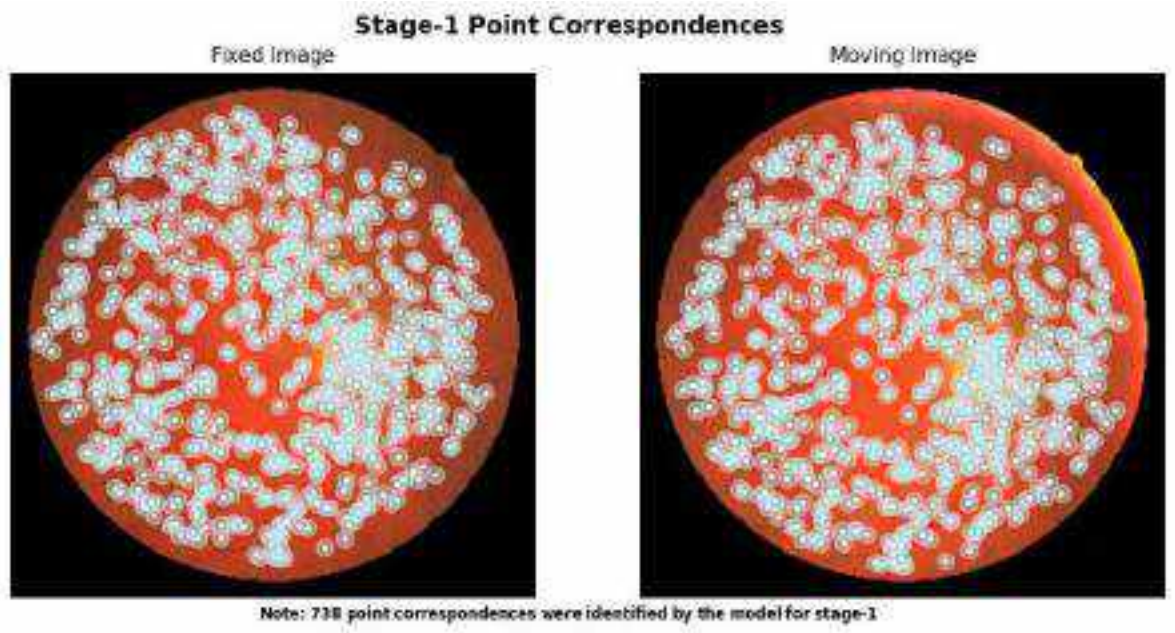

```
Homography Matrix:
[[ 9.98613401e-01  3.88173334e-02 -1.90892529e+01]
 [-3.91834464e-02  9.99832059e-01  4.38664598e+00]
 [-5.59209440e-07 -6.11612525e-07  1.00000000e+00]]
```


```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



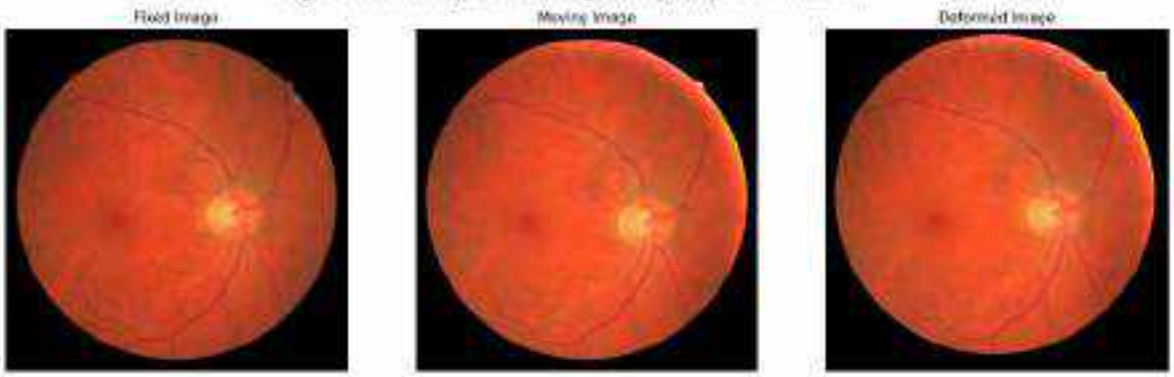



```
Mean Landmark Error for Case 46 Before Registration is 57.47792941069205 pixels
Mean Landmark Error for Case 46 After Registration is 1.2172086066855549 pixels
Case 47
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/548_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/548_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

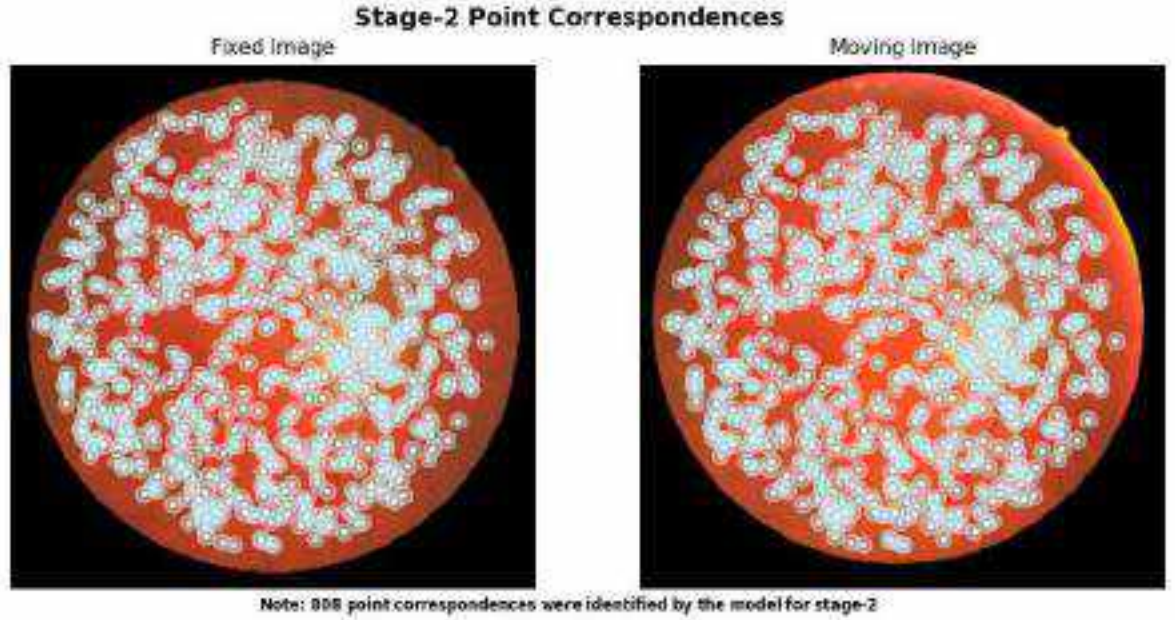


Note: 738 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.92811620e-01  5.23915197e-02 -3.01436419e+01]
 [-5.68256034e-02  9.94497634e-01  1.23051188e+01]
 [-5.17442982e-06 -2.07550956e-06  1.00000000e+00]]
```

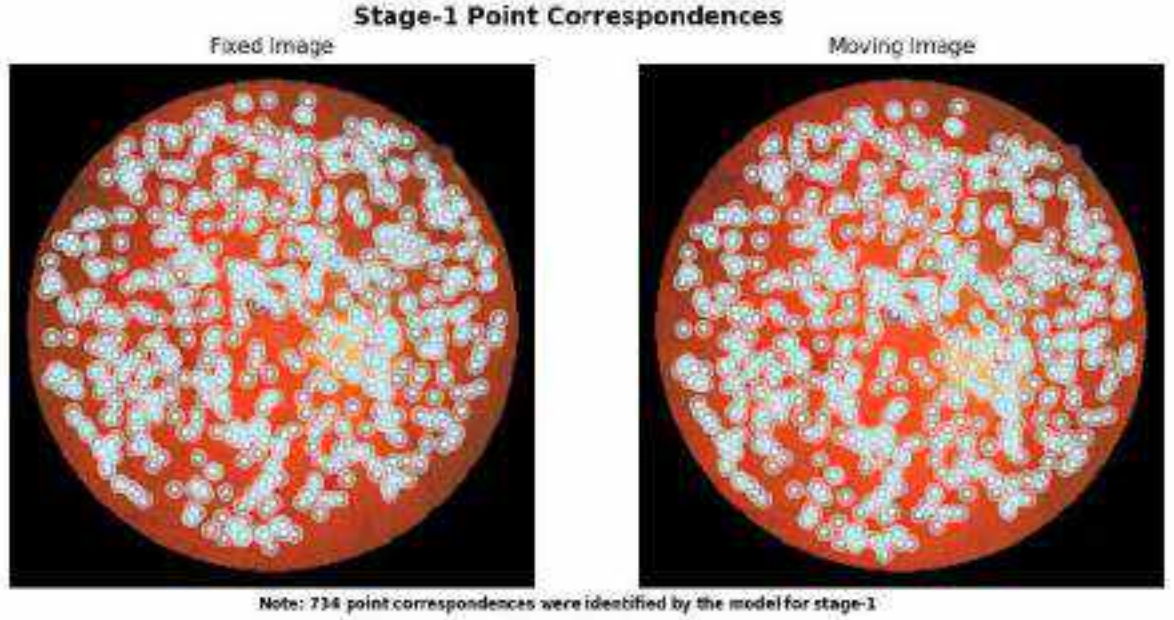



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

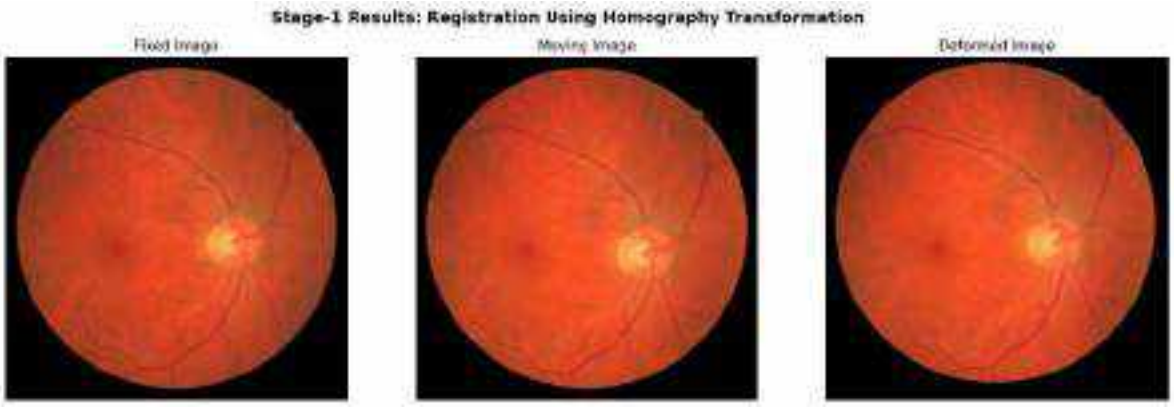


Note: 808 point correspondences were identified by the model for stage-2

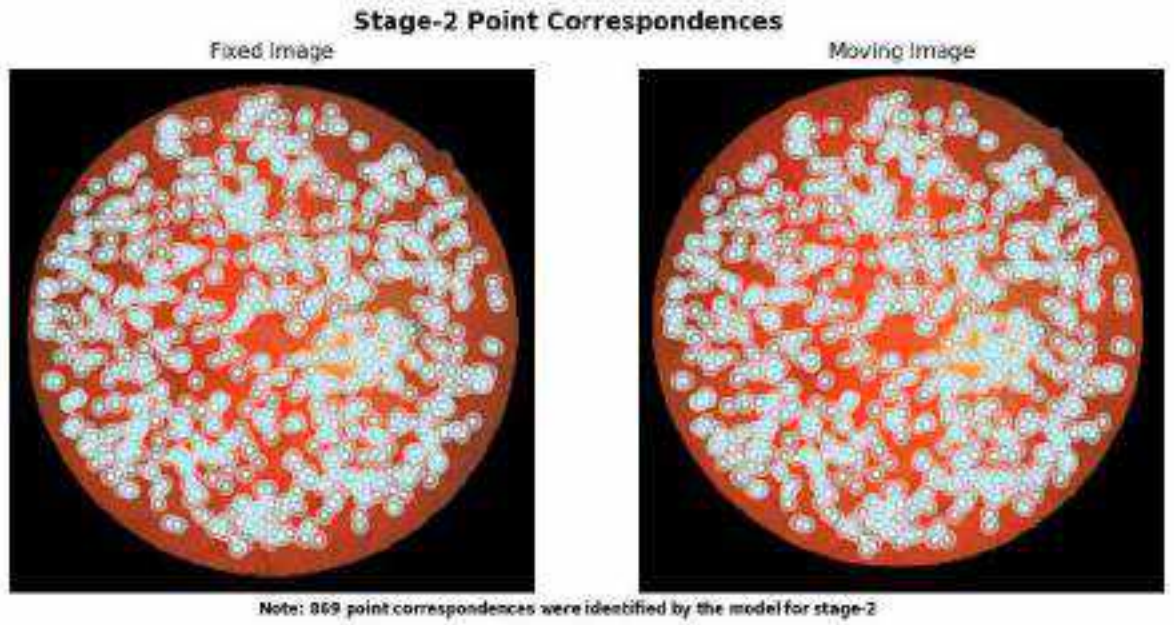

```
Mean Landmark Error for Case 47 Before Registration is 71.26386636411831 pixels
Mean Landmark Error for Case 47 After Registration is 1.1748116101667274 pixels
Case 48
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S49_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S49_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



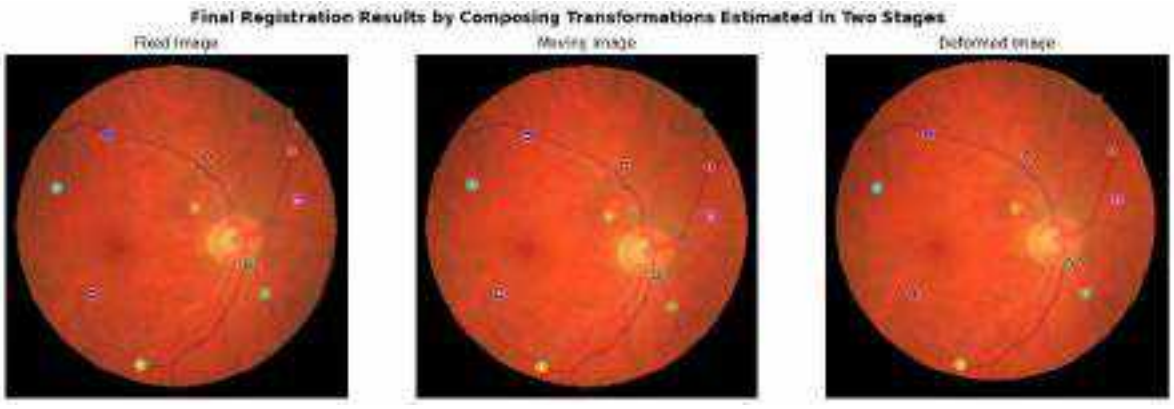

```
Homography Matrix:
[[ 9.89055318e-01  7.65500289e-02 -3.96784589e+01]
 [-8.22915368e-02  9.84692290e-01  2.46156121e+01]
 [-3.22646647e-06 -1.89944397e-05  1.00000000e+00]]
```



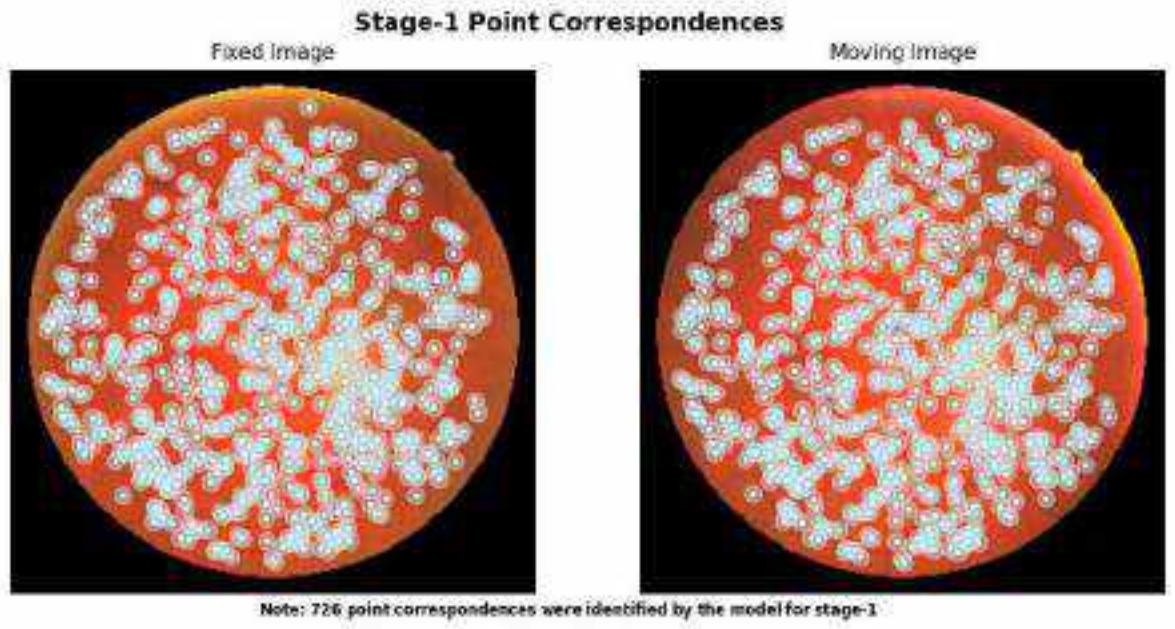

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 869 point correspondences were identified by the model for stage-2

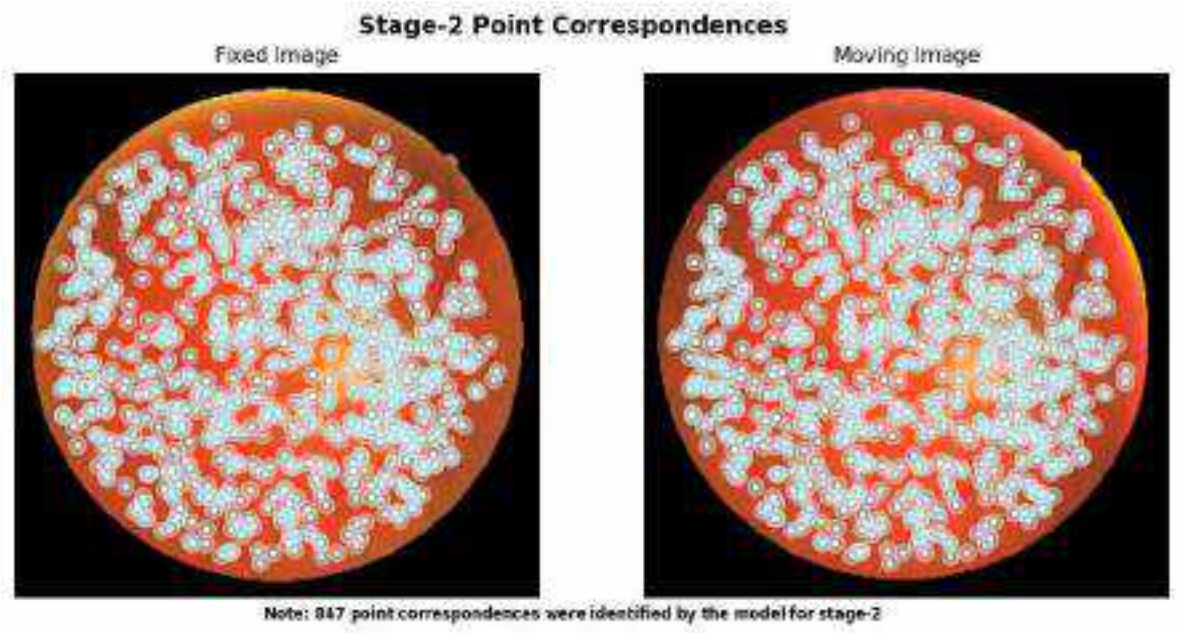


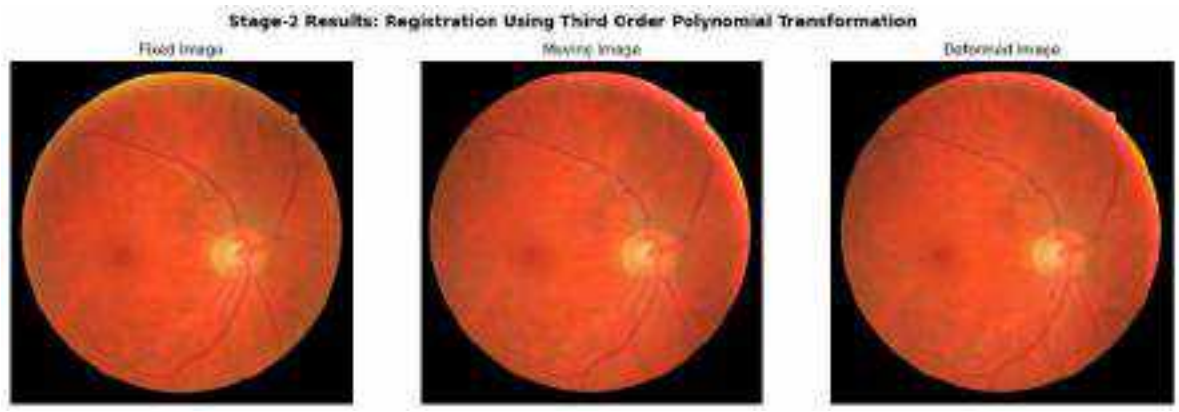
```
Mean Landmark Error for Case 48 Before Registration is 93.800402837337707 pixels
Mean Landmark Error for Case 48 After Registration is 1.5482851346311131 pixels
Case 49
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S50_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S50_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

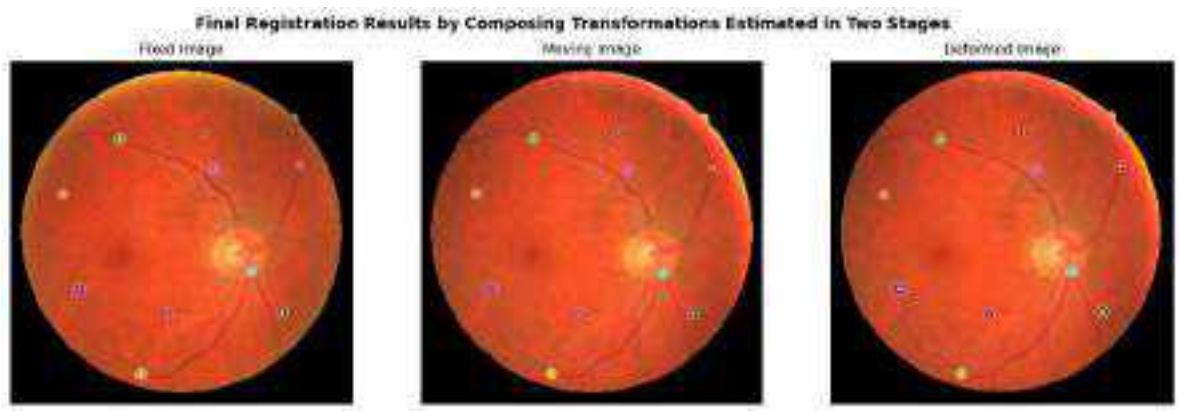



```
Homography Matrix:
[[ 9.94472111e-01  1.65922717e-02 -1.24186130e+01]
 [-1.73720631e-02  9.96925139e-01  6.61902330e+00]
 [-6.40824765e-06  5.18965541e-07  1.00000000e+00]]
```

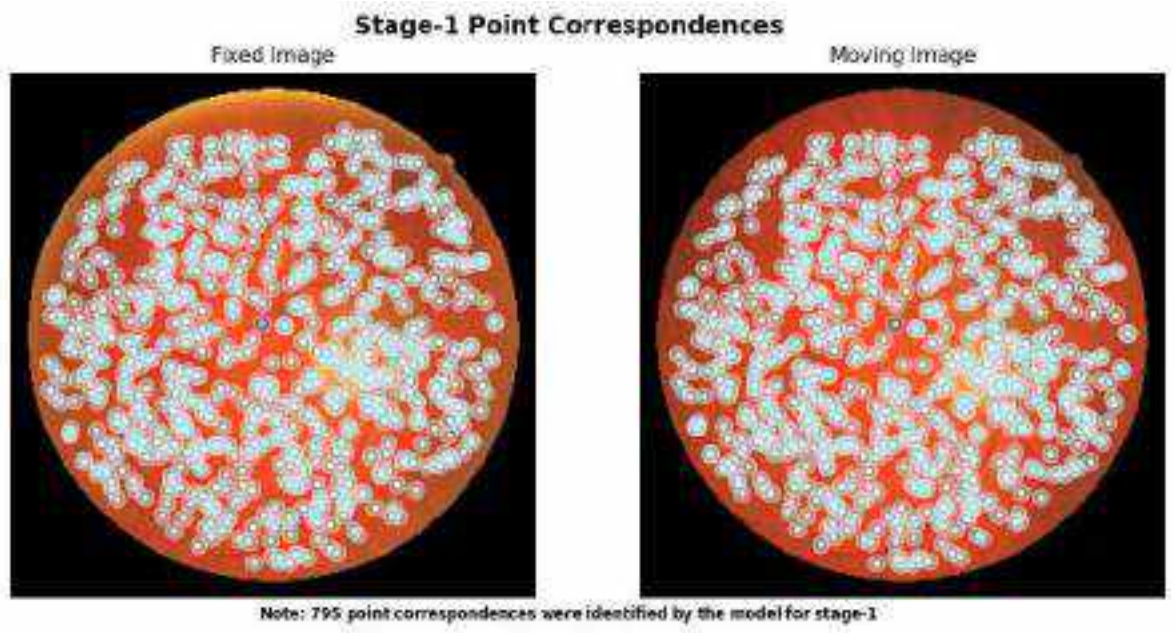

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





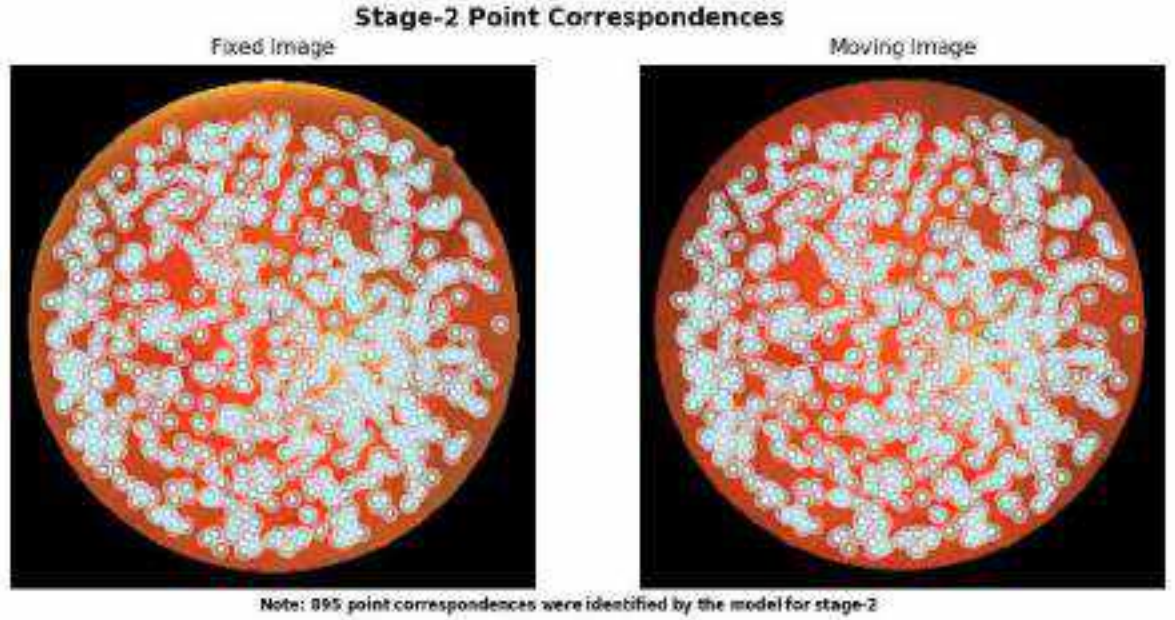



```
Mean Landmark Error for Case 49 Before Registration is 21.676323B0548B344 pixels
Mean Landmark Error for Case 49 After Registration is 1.3890338101911885 pixels
Case 50
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S51_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S51_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

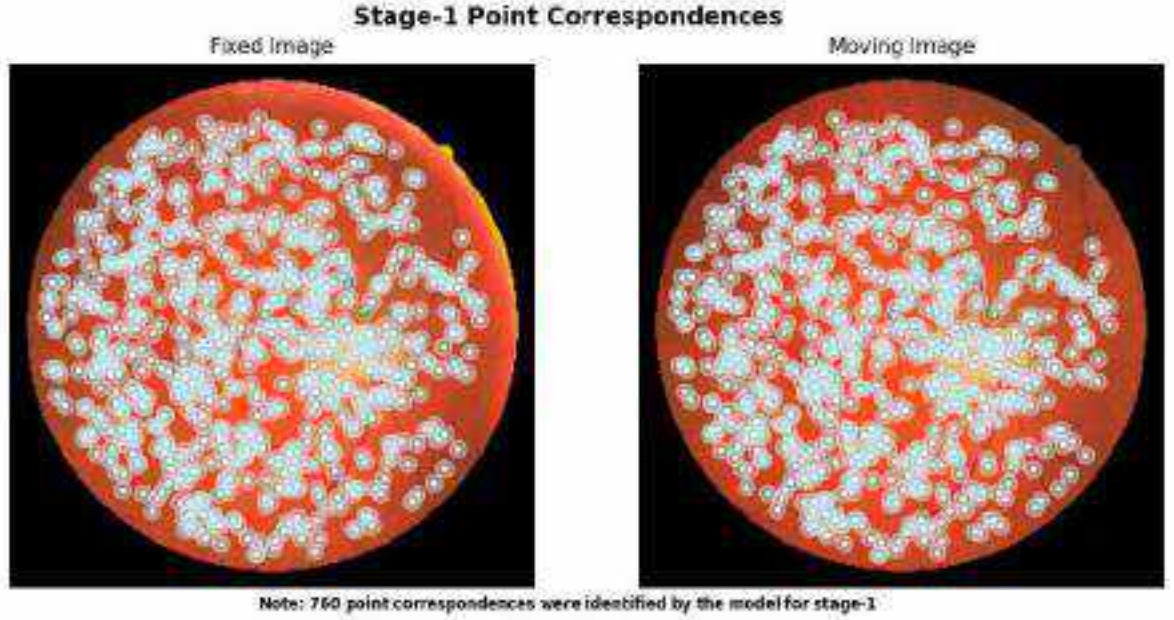



```
Homography Matrix:
[[ 9.91827994e-01  3.95771898e-02 -2.15315161e+01]
 [-4.39986838e-02  9.90986932e-01  1.81039956e+01]
 [-5.11684232e-06 -5.77047784e-06  1.00000000e+00]]
```

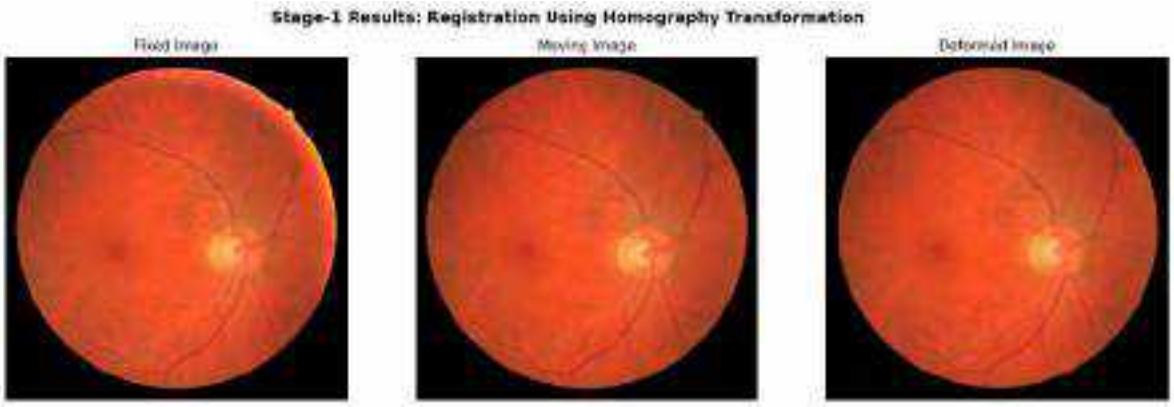

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



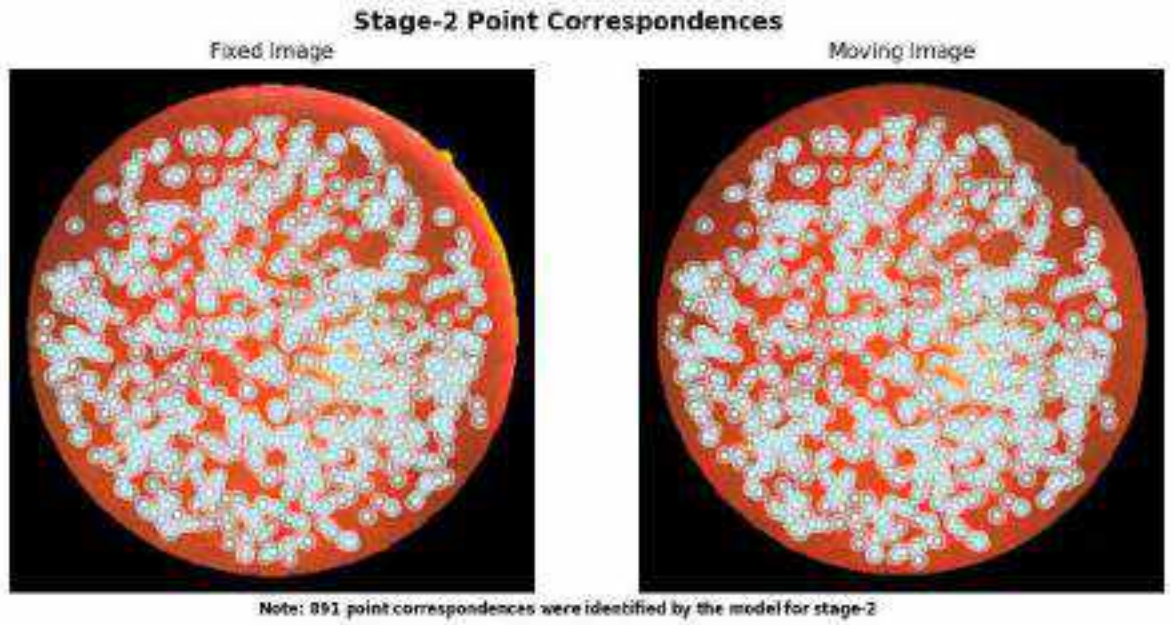

```
Mean Landmark Error for Case 50 Before Registration is 40.57071134229913 pixels
Mean Landmark Error for Case 50 After Registration is 0.8210026517648036 pixels
Case 51
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S52_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S52_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



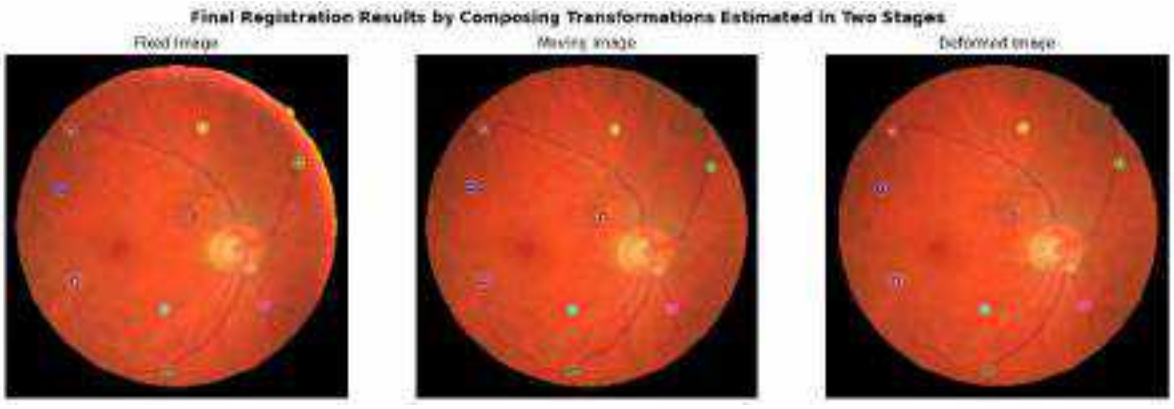

```
Homography Matrix:
[[ 1.00010347e+00  2.44253987e-02 -1.00955433e+01]
 [-2.47190371e-02  9.97276512e-01  1.00033691e+01]
 [ 2.71993824e-06 -3.93741701e-06  1.00000000e+00]]
```



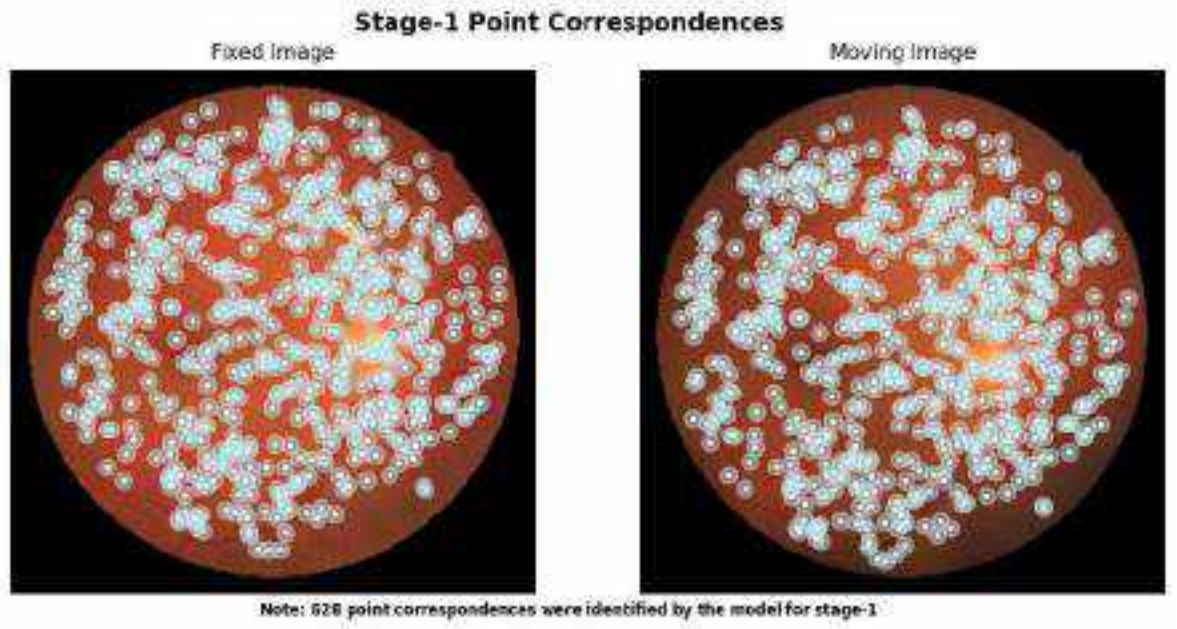

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



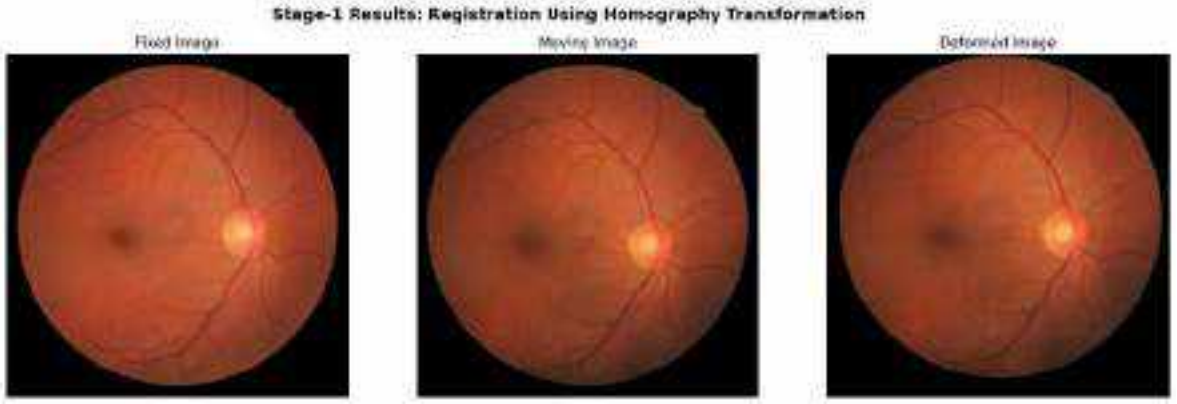
Note: 891 point correspondences were identified by the model for stage-2



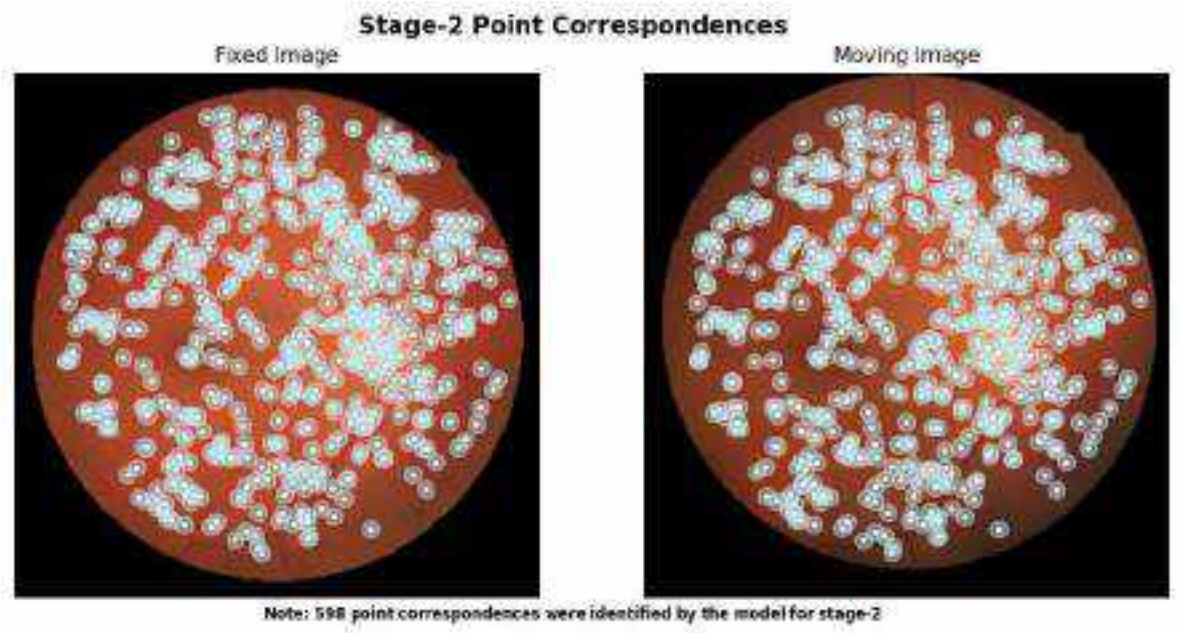
```
Mean Landmark Error for Case 51 Before Registration is 26.8212334238114 35 pixels
Mean Landmark Error for Case 51 After Registration is 1.3429284304496865 pixels
Case 52
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S53_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S53_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

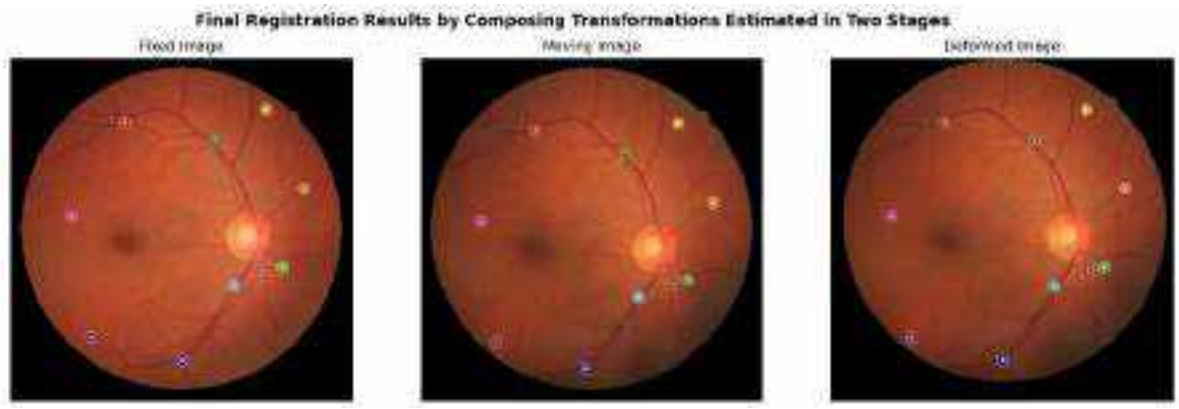



```
Homography Matrix:
[[ 1.00541632e+00  3.26601026e-02 -1.35625169e+01]
 [-3.52160034e-02  9.95414806e-01 -9.81338423e+00]
 [ 3.98824273e-06 -1.40368105e-05  1.00000000e+00]]
```

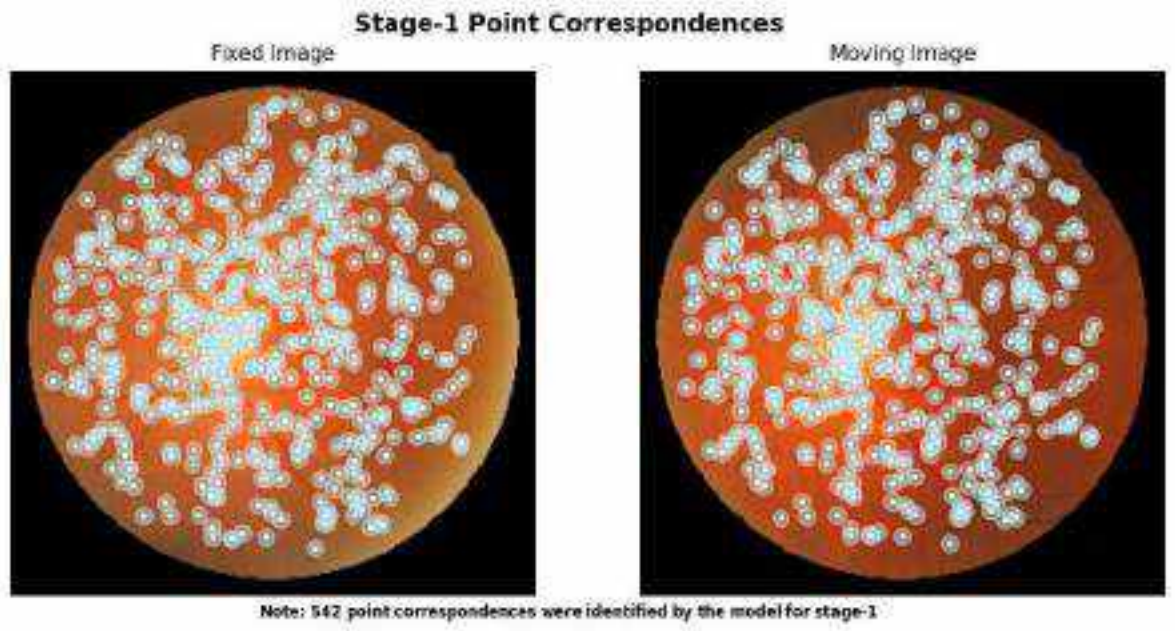



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





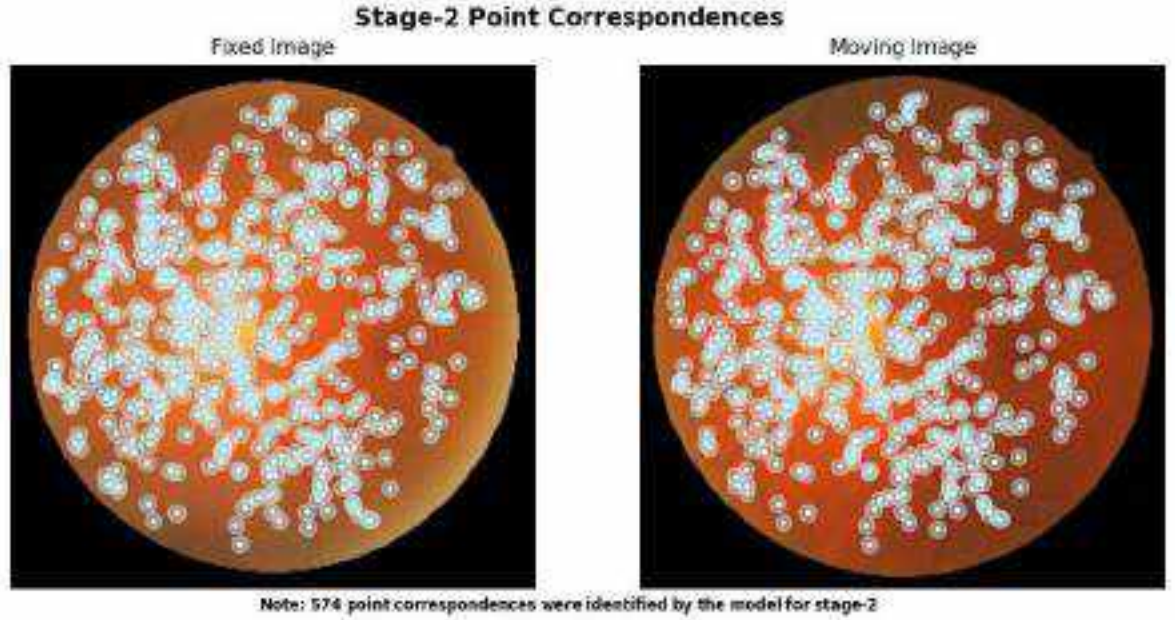

```
Mean Landmark Error for Case 52 Before Registration is 95.00923652928002 pixels
Mean Landmark Error for Case 52 After Registration is 4.250439725493935 pixels
Case 53
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S54_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S54_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



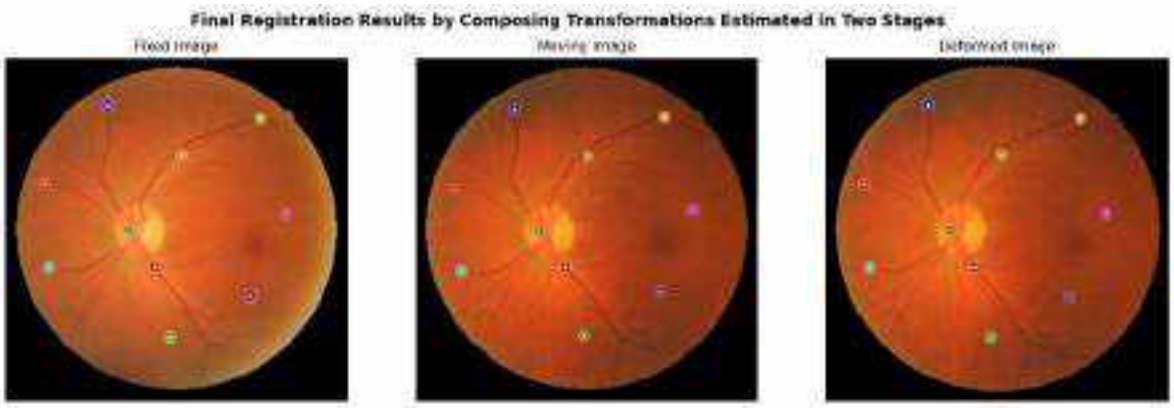
Note: 542 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 1.02105790e+00 -3.51158046e-02  9.43693636e+00]
 [ 3.62644213e-02  1.01949328e+00 -2.43615380e+01]
 [ 5.07582023e-06  1.25596617e-07  1.00000000e+00]]
```

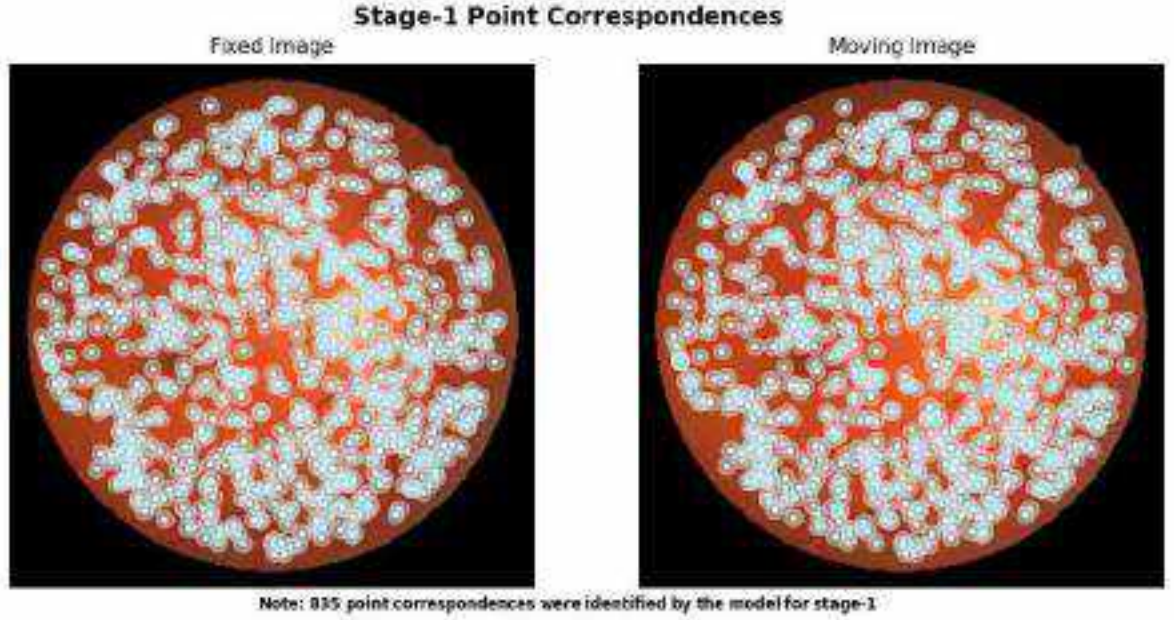

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



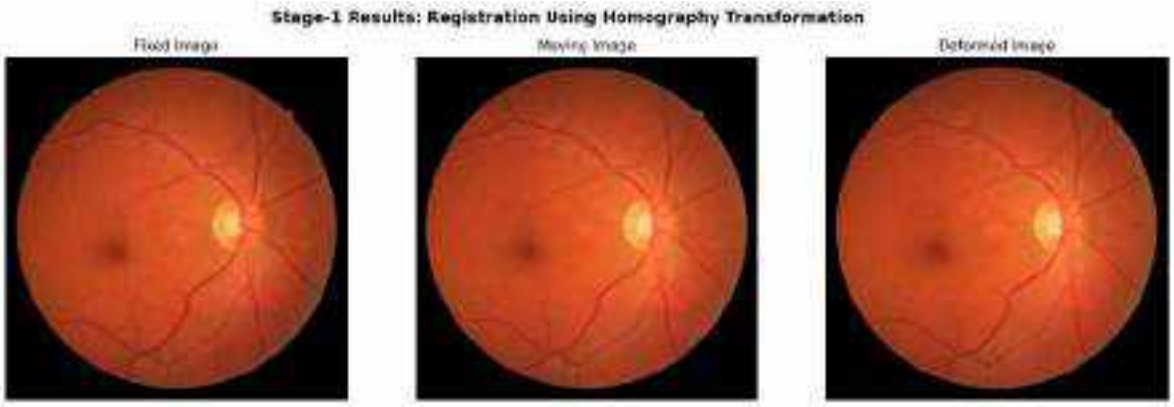
Note: 574 point correspondences were identified by the model for stage-2



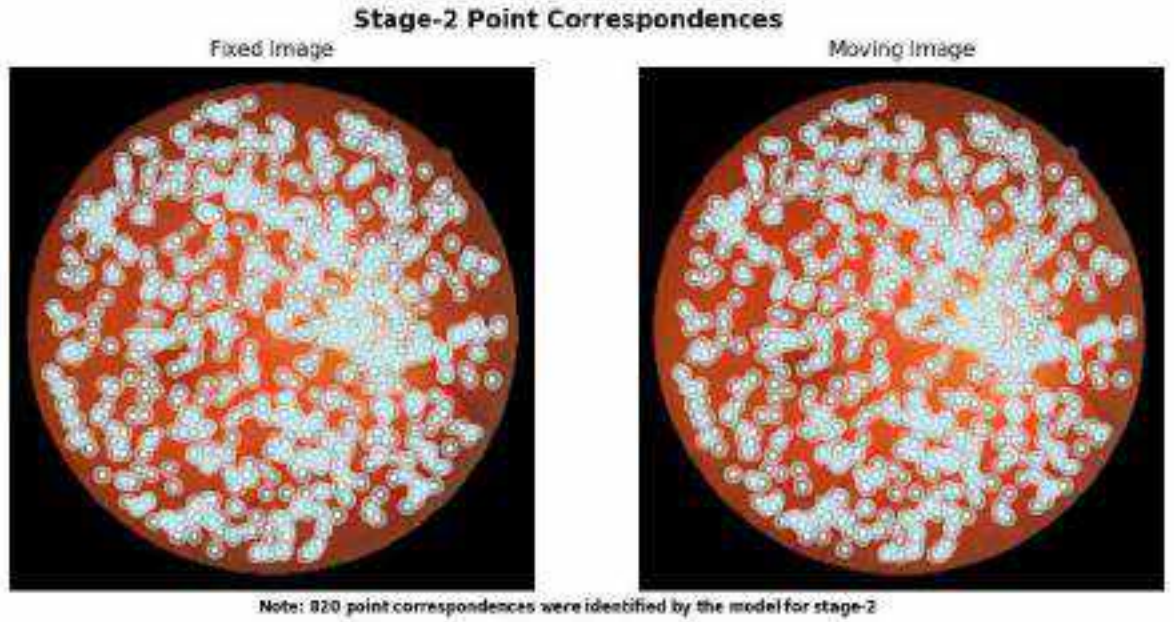

```
Mean Landmark Error for Case 53 Before Registration is 34.209616524430565 pixels
Mean Landmark Error for Case 53 After Registration is 2.2072823658503121 pixels
Case 54
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S55_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S55_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



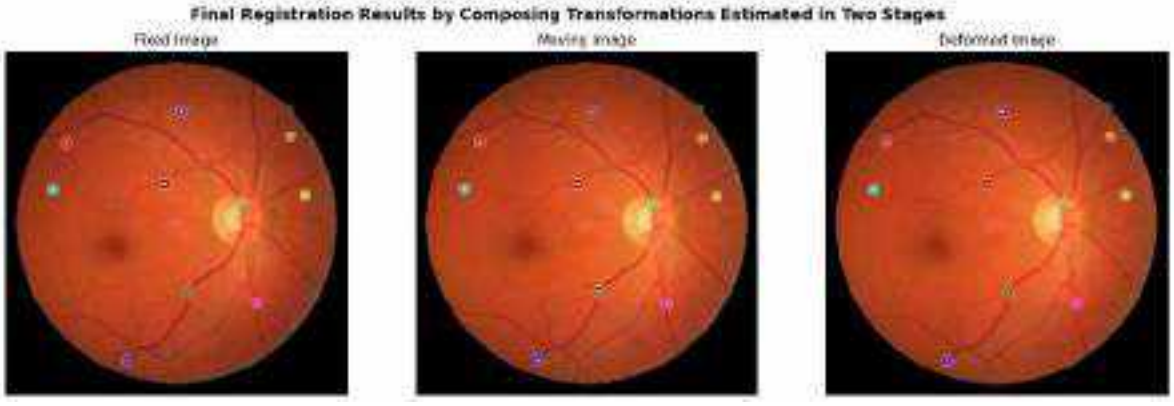

```
Homography Matrix:
[[ 9.99068178e-01  1.03336400e-02 -8.47874704e+00]
 [-9.71096812e-03  1.00437853e+00  1.64228260e+00]
 [-4.78167229e-06  2.44627358e-06  1.00000000e+00]]
```



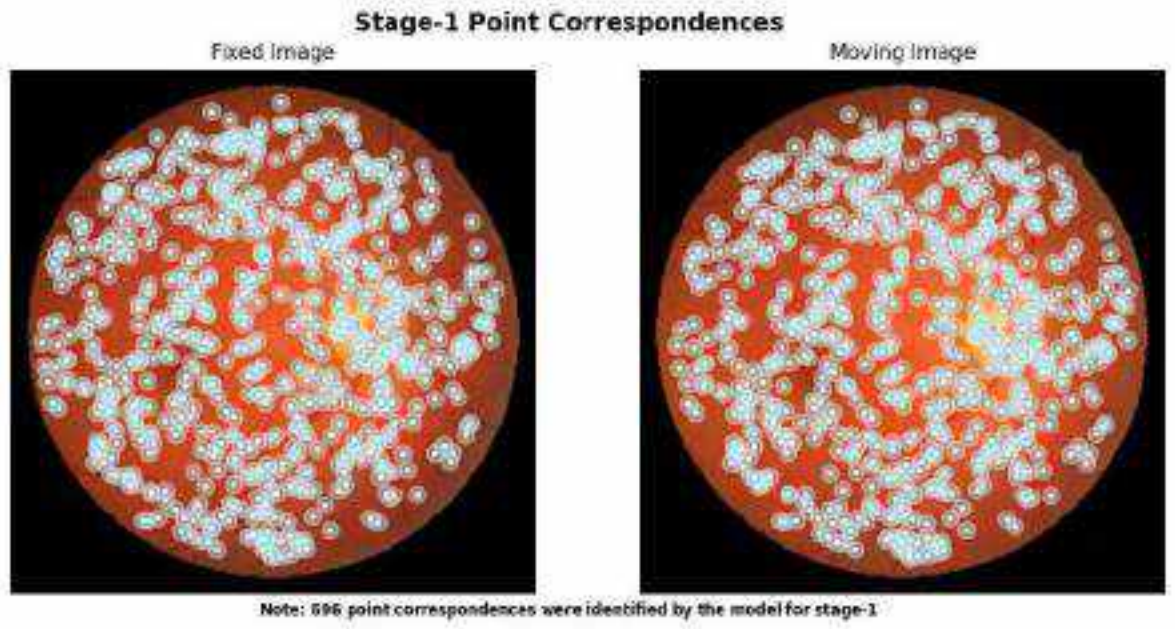

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



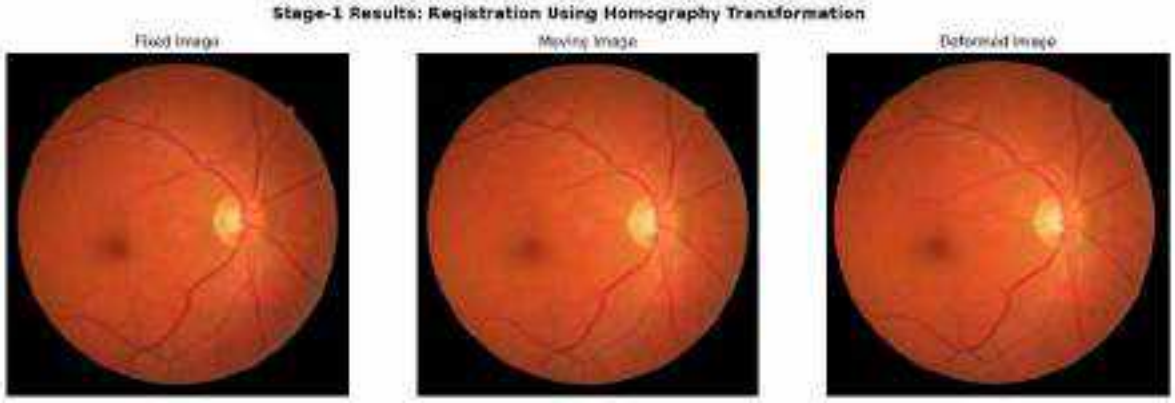
Note: 820 point correspondences were identified by the model for stage-2



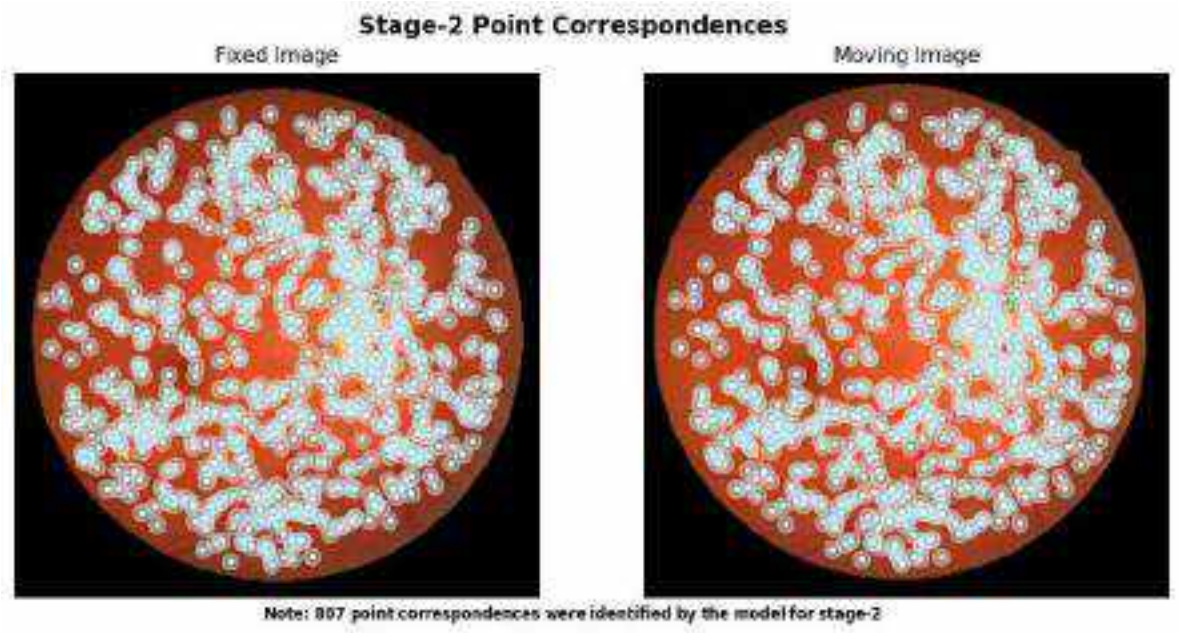
```
Mean Landmark Error for Case 54 Before Registration is 14.926373679223905 pixels
Mean Landmark Error for Case 54 After Registration is 1.2968087541644448 pixels
Case 55
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S56_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S56_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

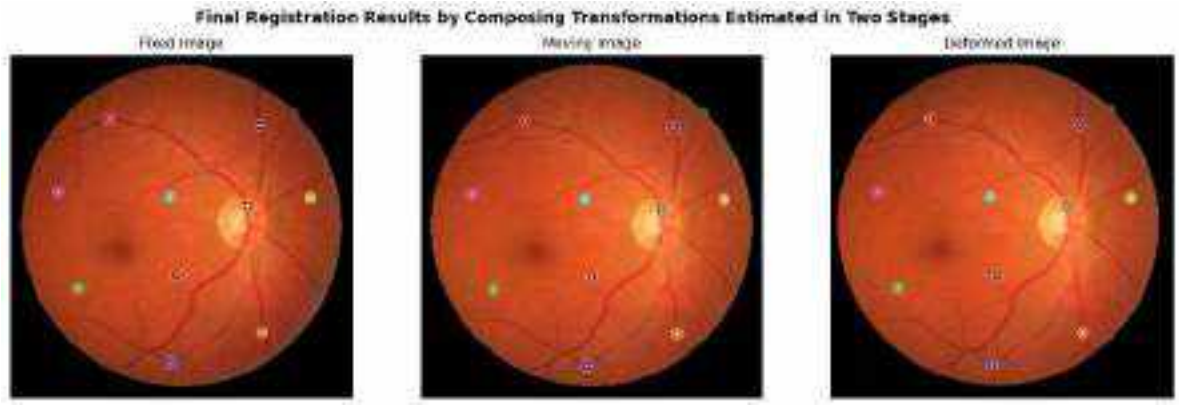



```
Homography Matrix:
[[ 9.92037589e-01 -5.91202181e-04 -1.21352327e+01]
 [-4.43758982e-03  1.00303034e+00 -6.58734560e+00]
 [-1.49187879e-05  3.11208420e-06  1.00000000e+00]]
```

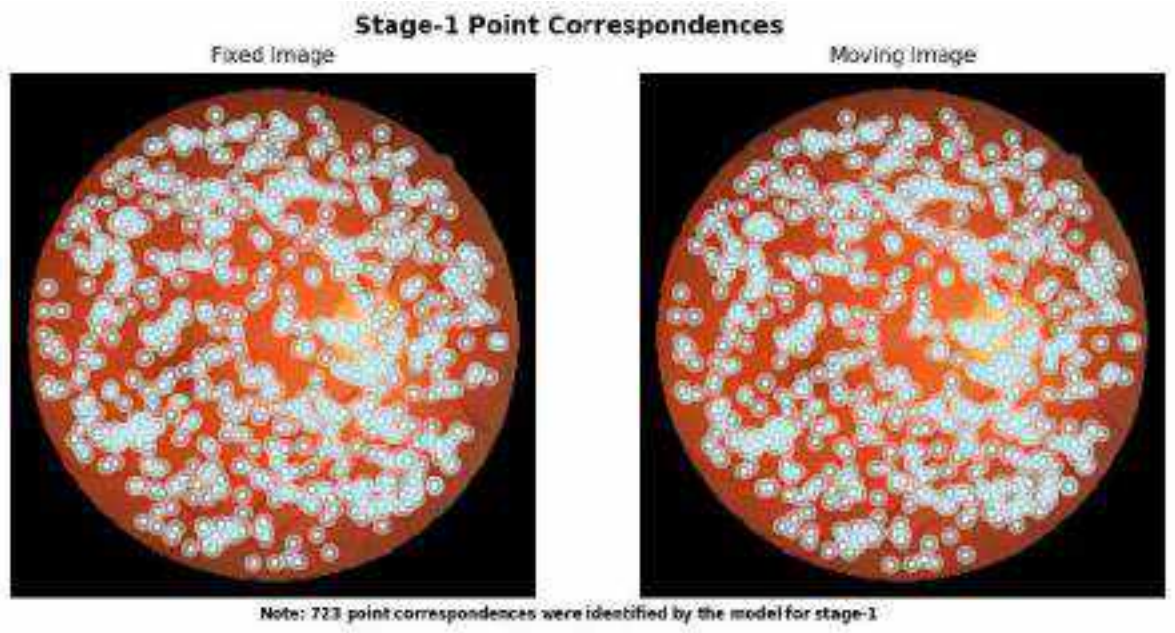



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

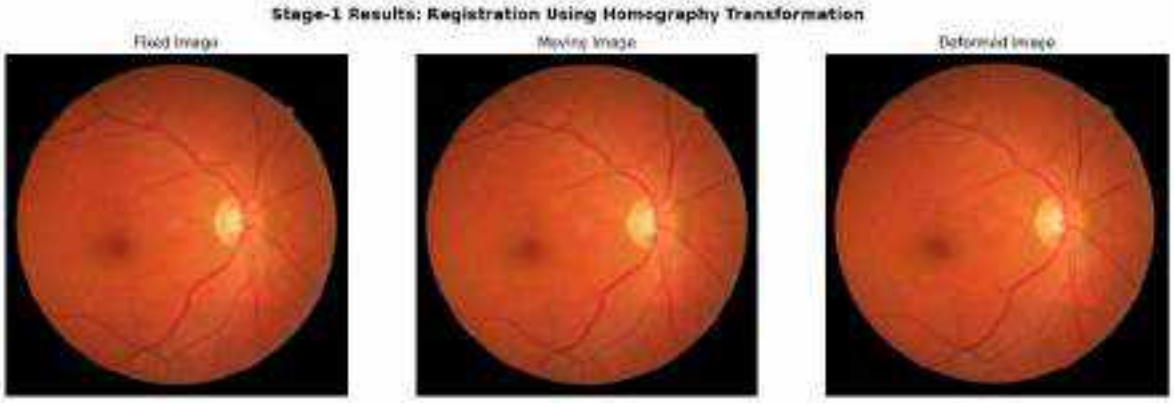




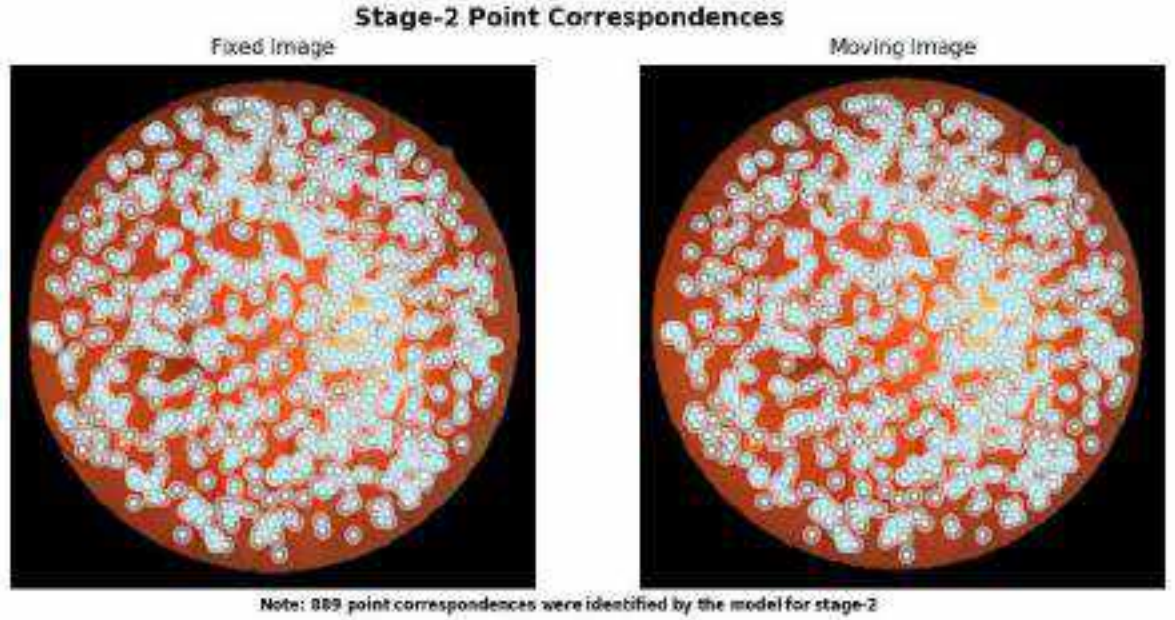

```
Mean Landmark Error for Case 55 Before Registration is 42.68941689638 74 pixels
Mean Landmark Error for Case 55 After Registration is 1.9337895981366646 pixels
Case 56
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S57_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S57_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

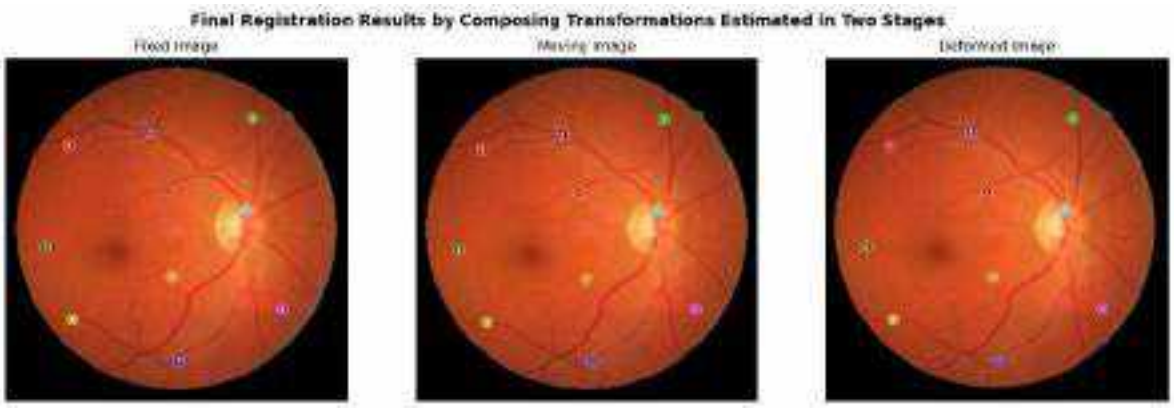



```
Homography Matrix:
[[ 9.91281956e-01 -1.46503739e-02 -5.38005966e-01]
 [ 9.59629866e-03  9.92529844e-01 -8.49909688e+00]
 [-7.18054905e-06 -4.77707442e-06  1.00000000e+00]]
```

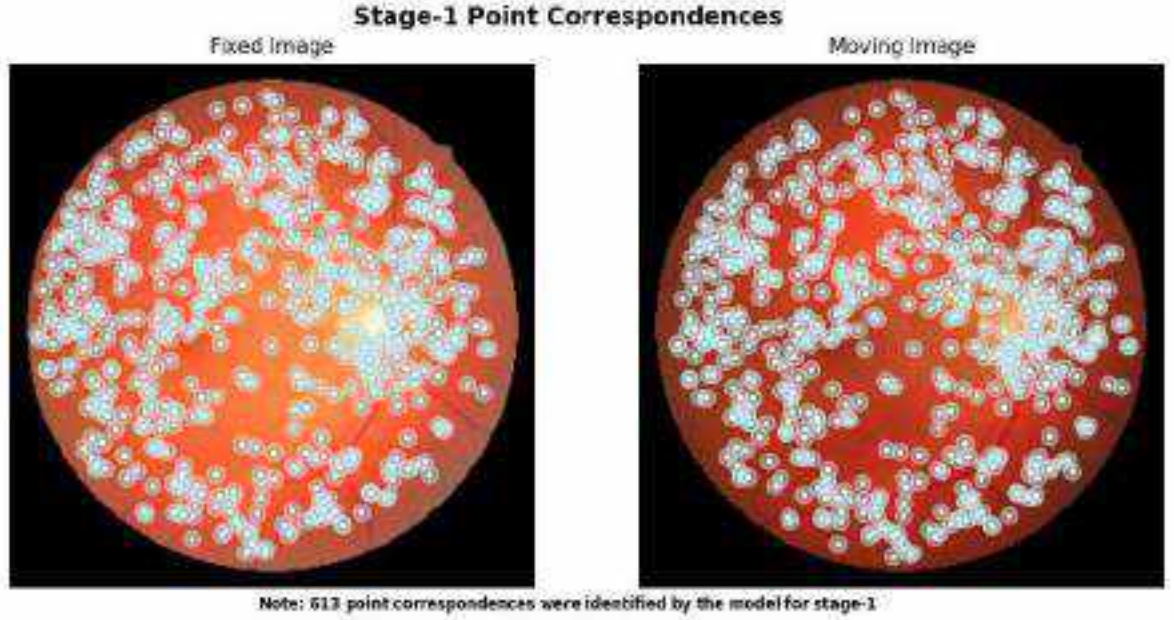



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

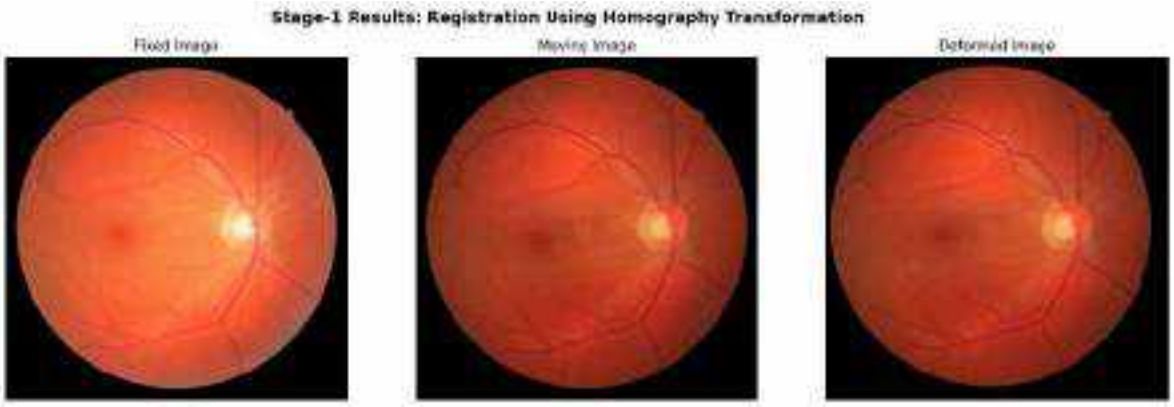




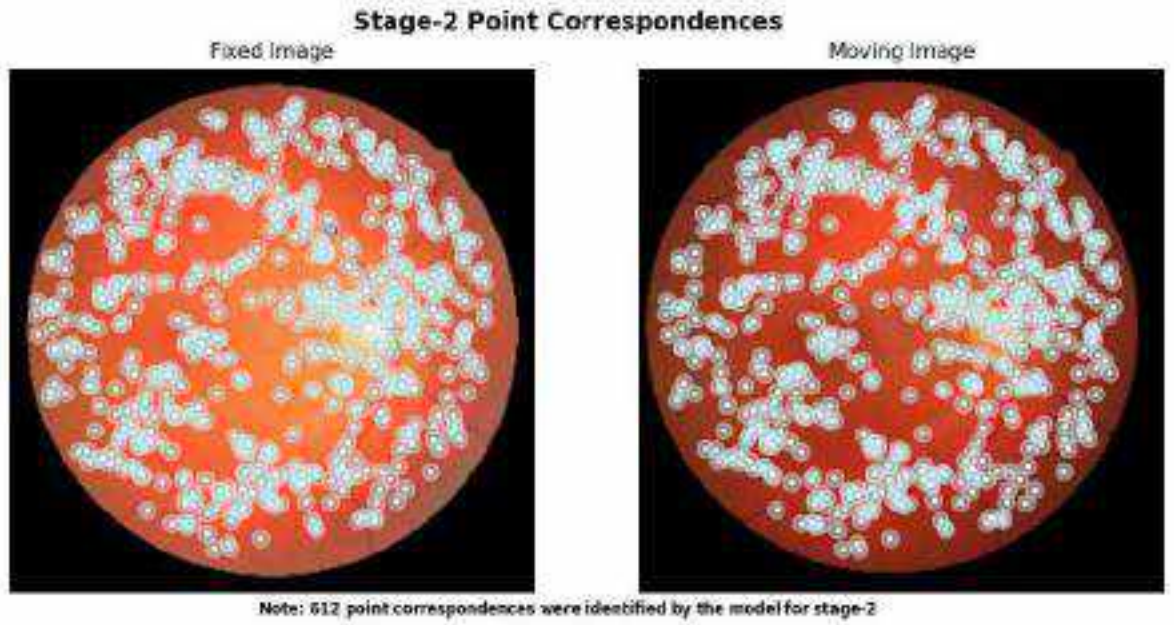

```
Mean Landmark Error for Case 56 Before Registration is 32.93825014160733 pixels
Mean Landmark Error for Case 56 After Registration is 1.1984725793458006 pixels
Case 57
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S58_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S58_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



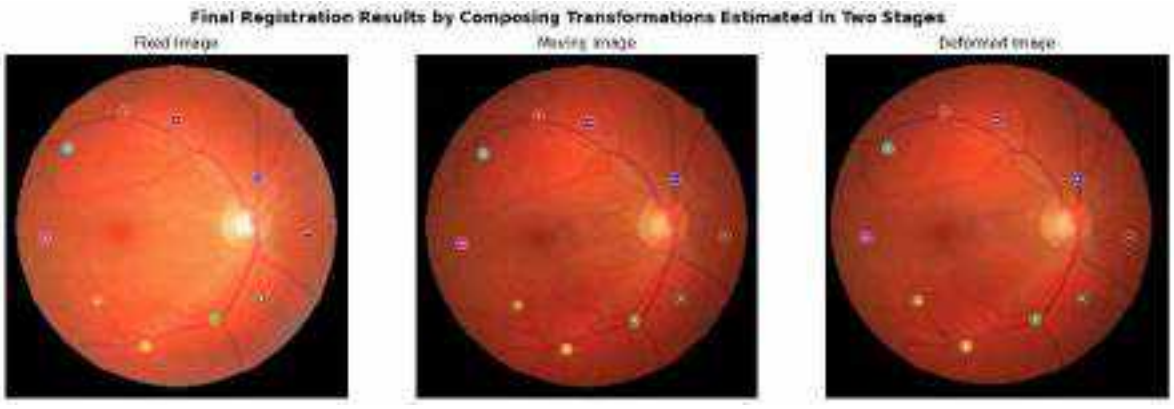

```
Homography Matrix:
[[ 9.84042716e-01 -3.24257031e-02 -2.33998214e+00]
 [ 2.23073011e-02  9.86402152e-01 -1.58939986e+01]
 [-1.48885479e-05 -8.12227516e-06  1.00000000e+00]]
```



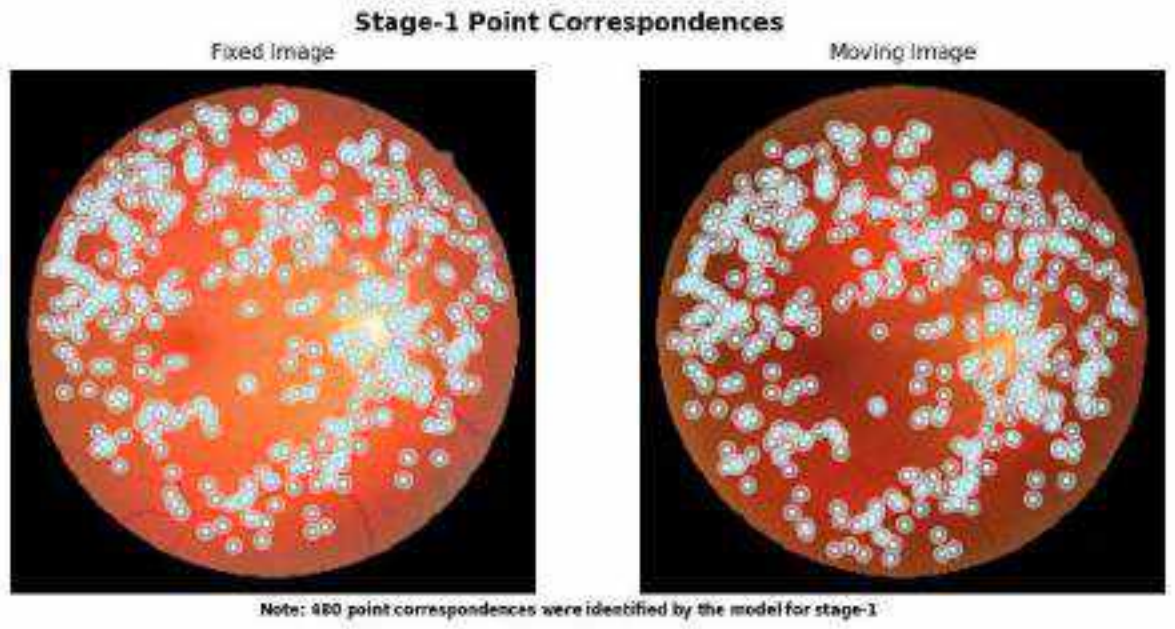

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



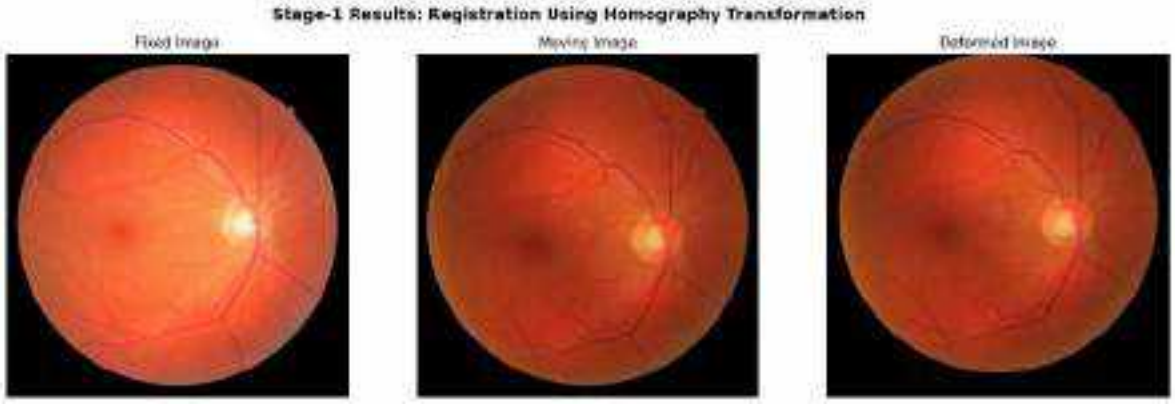
Note: 612 point correspondences were identified by the model for stage-2



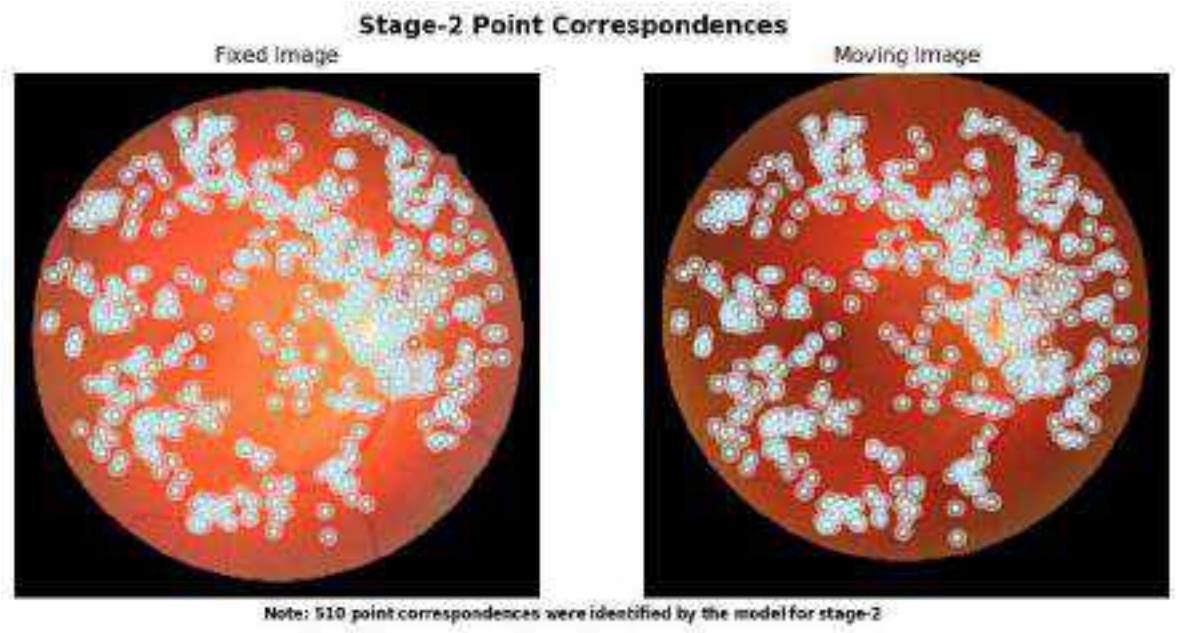
```
Mean Landmark Error for Case 57 Before Registration is 68.71656249979627 pixels
Mean Landmark Error for Case 57 After Registration is 1.8427226416910933 pixels
Case 58
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S59_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S59_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

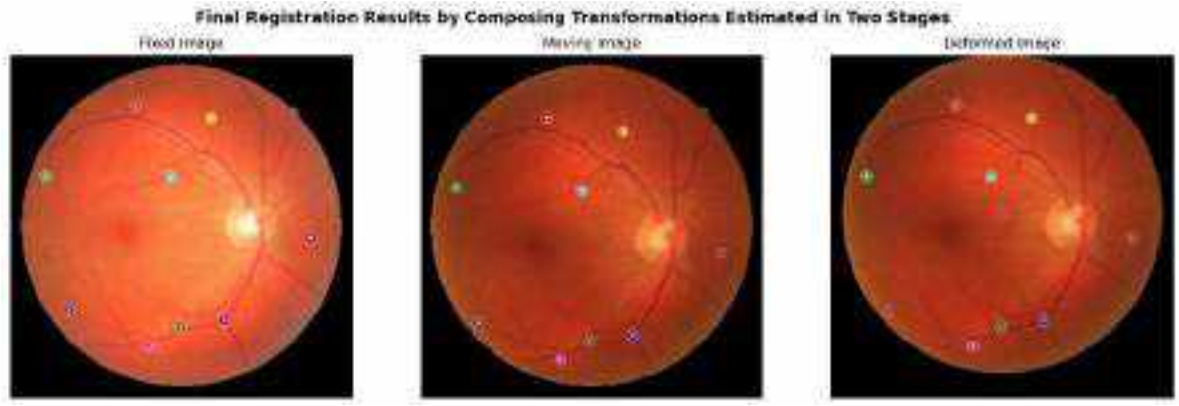



```
Homography Matrix:
[[ 9.82156830e-01  2.90610369e-04  1.08588672e+00]
 [-1.47013954e-02  9.73207234e-01 -2.40269469e+01]
 [-3.83186477e-06 -2.31694690e-05  1.00000000e+00]]
```

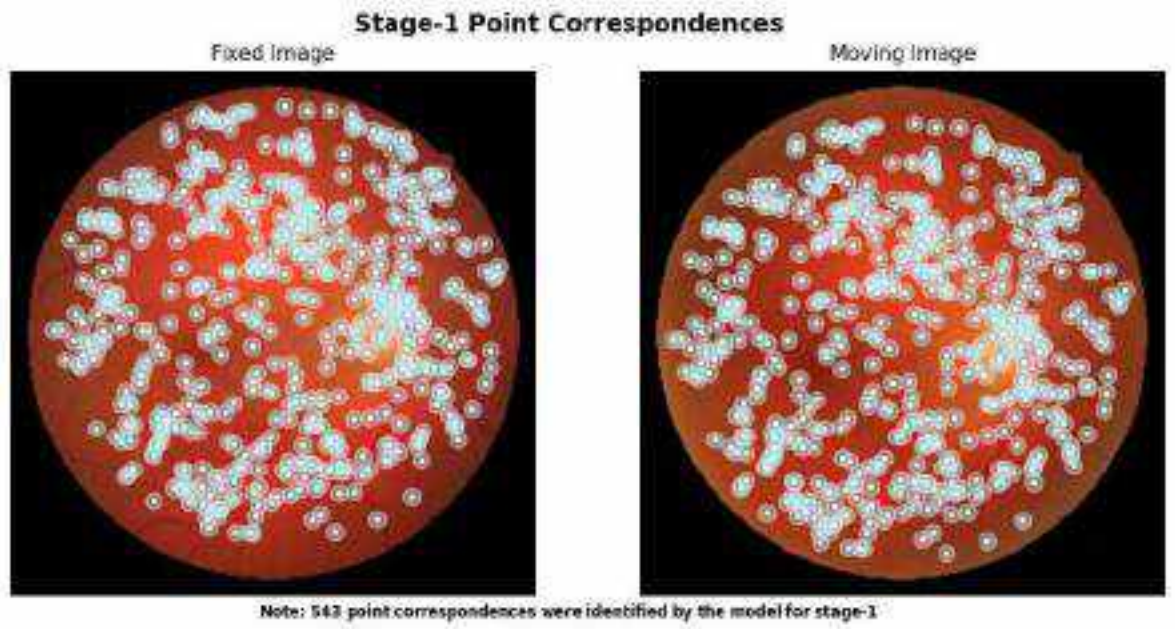



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



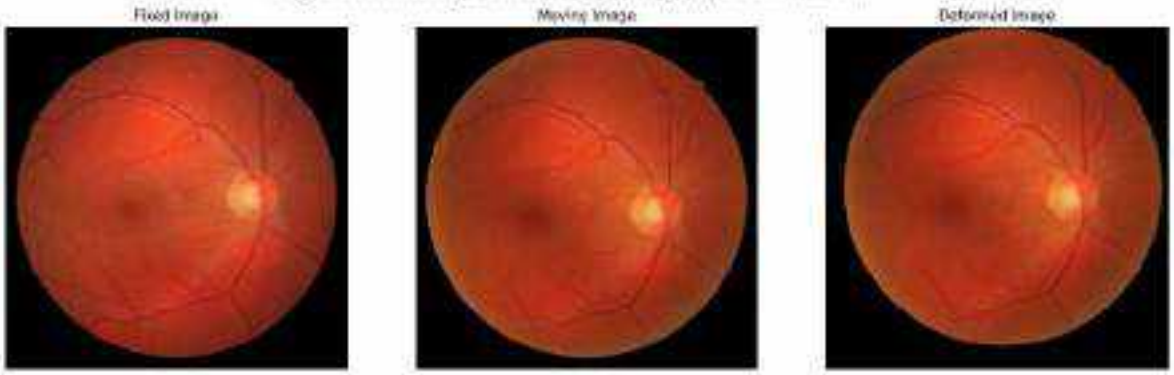



```
Mean Landmark Error for Case 58 Before Registration is 118.03170816521984 pixels
Mean Landmark Error for Case 58 After Registration is 2.5201098868852948 pixels
Case 59
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S60_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S60_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

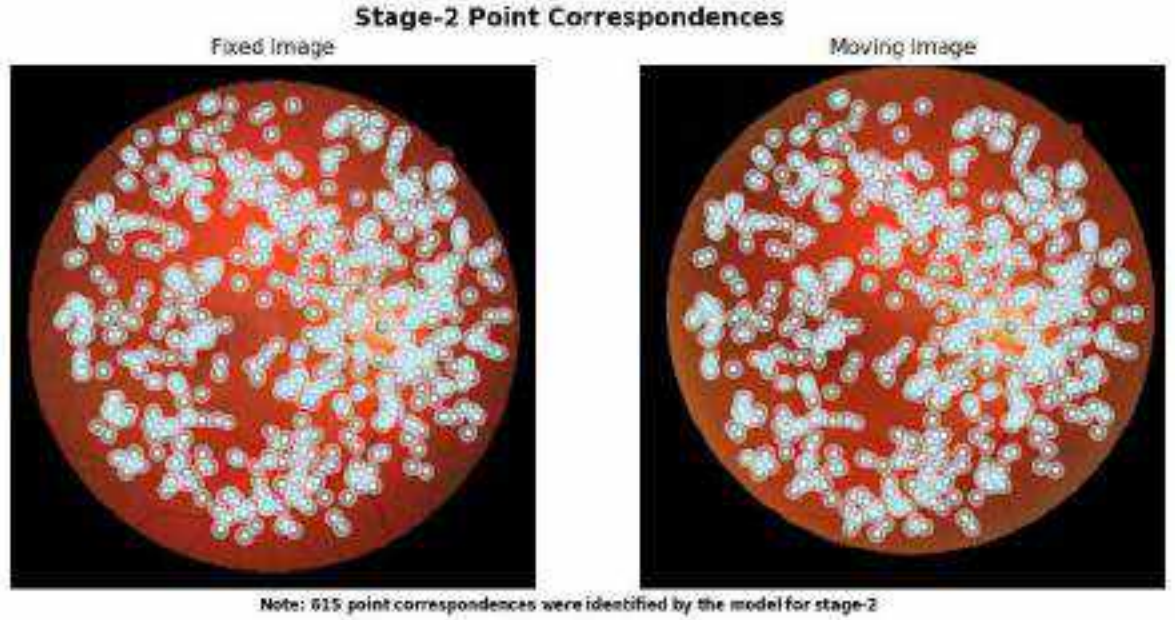


Note: 543 point correspondences were identified by the model for stage-1

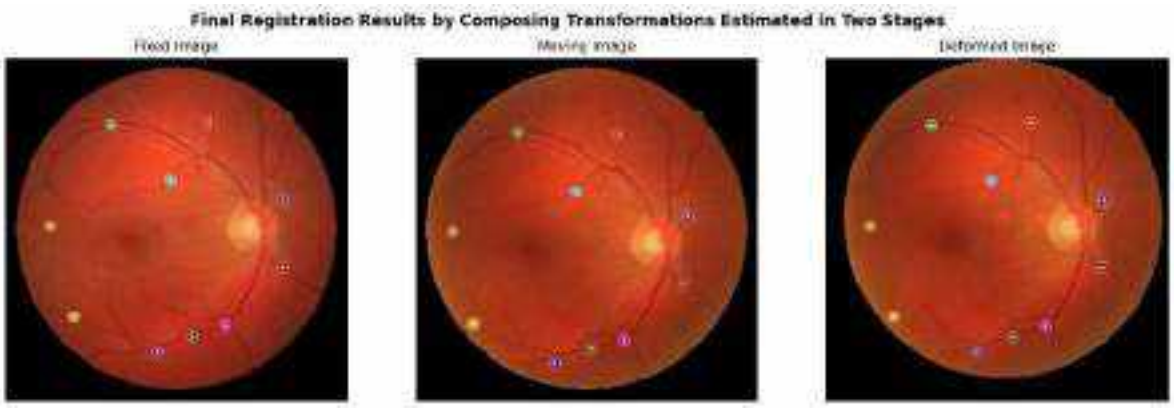
```
Homography Matrix:
[[ 1.00025210e+00  3.76598823e-02  4.84758888e-01]
 [-3.51707924e-02  9.90071310e-01 -1.03478880e+01]
 [ 1.18789443e-05 -1.00297309e-05  1.00000000e+00]]
```



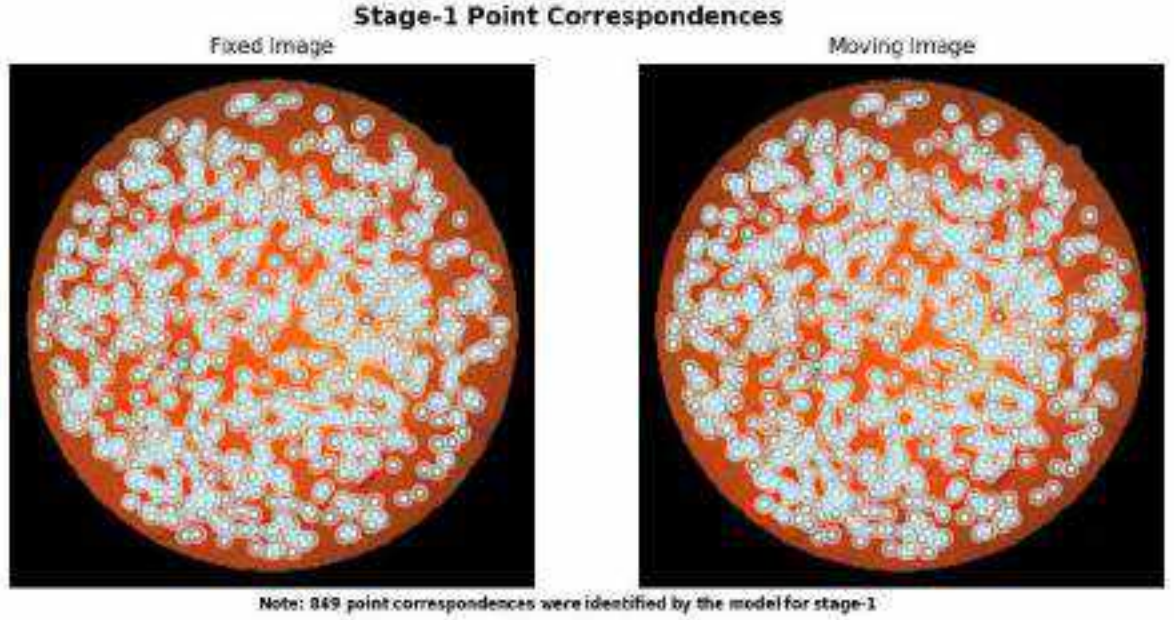

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



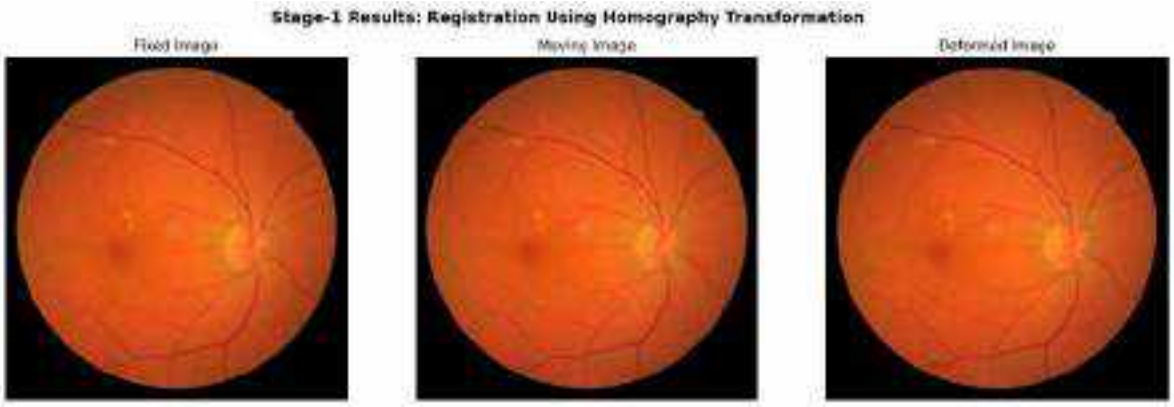
Note: 615 point correspondences were identified by the model for stage-2



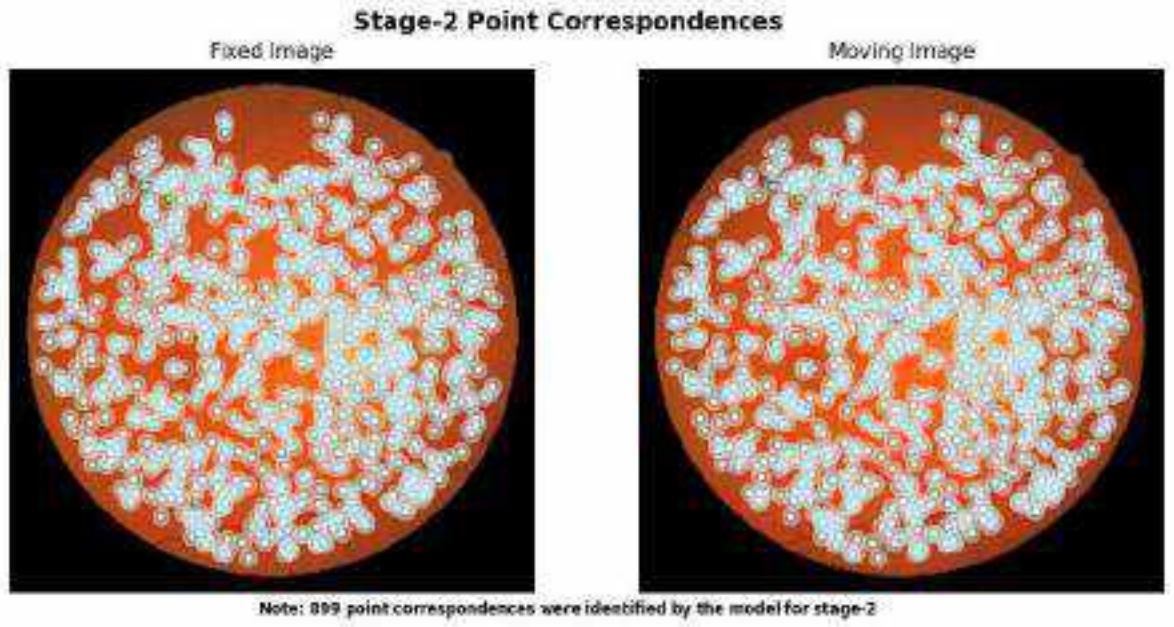

```
Mean Landmark Error for Case 59 Before Registration is 119.85722120134726 pixels
Mean Landmark Error for Case 59 After Registration is 2.257391279207949 pixels
Case 60
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S61_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S61_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



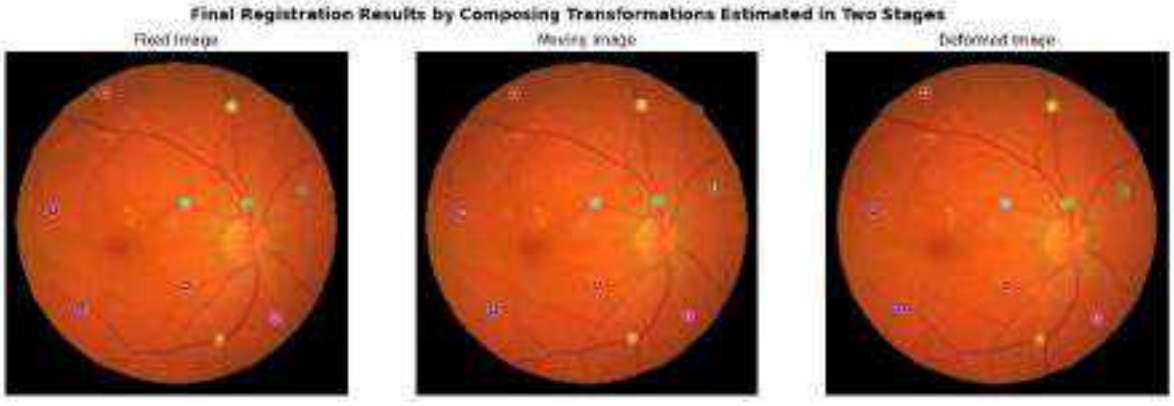

```
Homography Matrix:
[[ 9.98712987e-01 -1.43575835e-02  4.95705696e+00]
 [ 1.50763186e-02  1.00130745e+00 -6.24105538e+00]
 [-2.09747630e-06  3.26585248e-06  1.00000000e+00]]
```



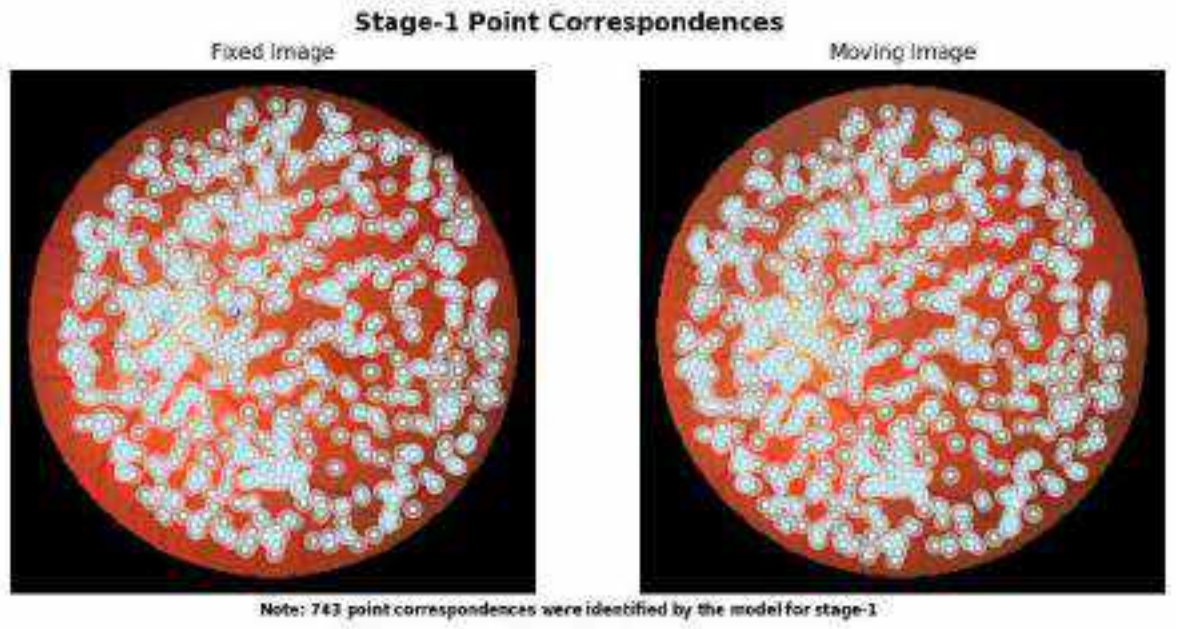

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



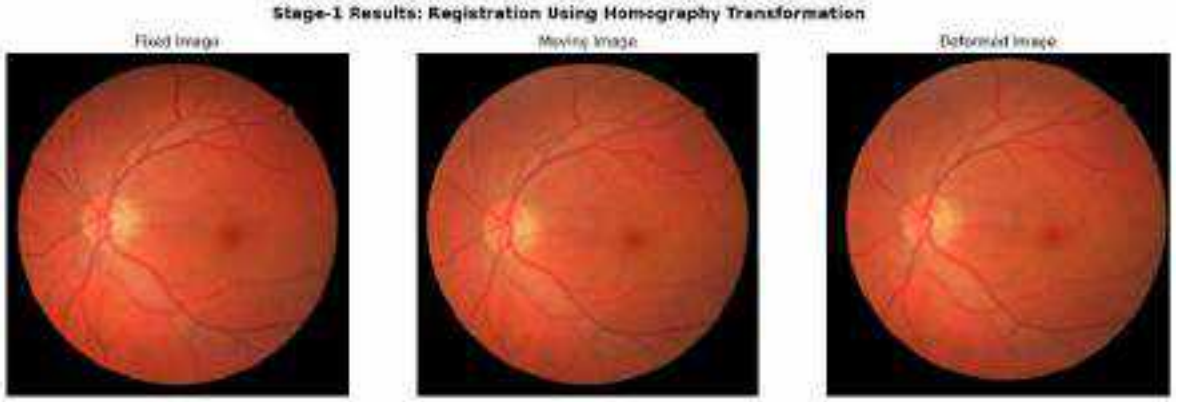
Note: 899 point correspondences were identified by the model for stage-2



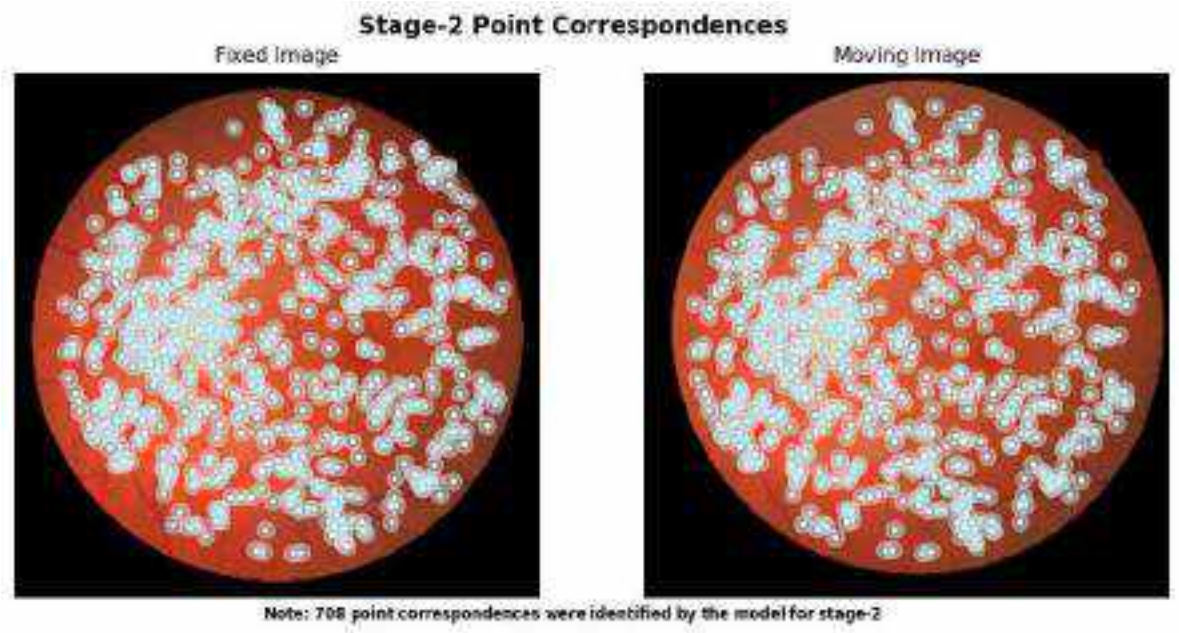
```
Mean Landmark Error for Case 60 Before Registration is 18.87545098516224 pixels
Mean Landmark Error for Case 60 After Registration is 1.198719361586099 pixels
Case 61
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S62_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S62_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

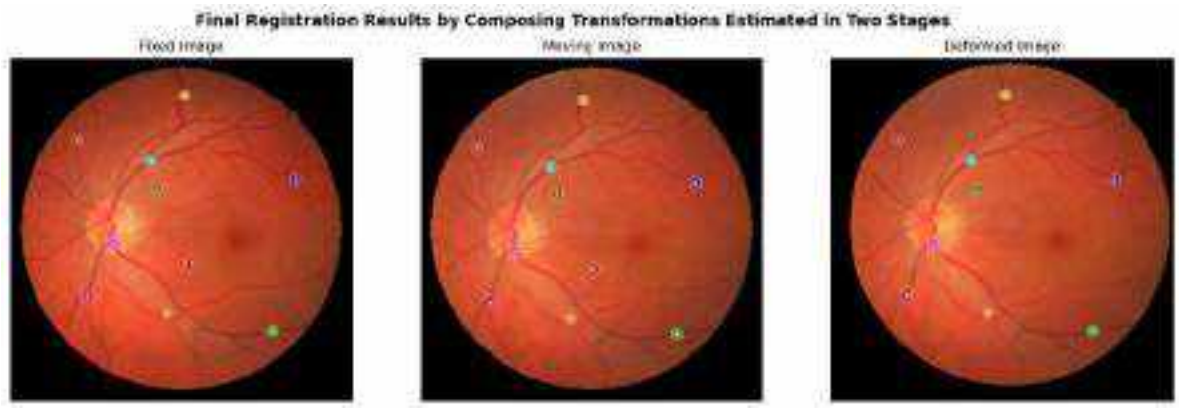



```
Homography Matrix:
[[ 1.01049138e+00 -2.80170098e-02  3.32075712e+01]
 [ 2.92759824e-02  1.00295053e+00 -2.75994658e+01]
 [ 1.49694360e-05 -7.81994727e-06  1.00000000e+00]]
```

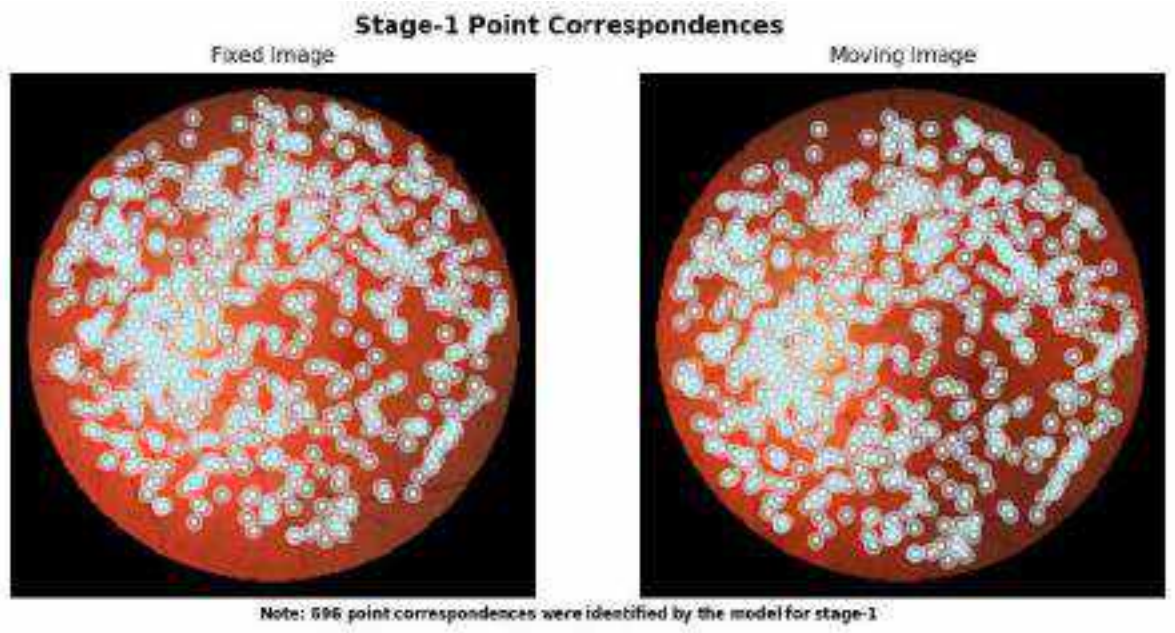



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





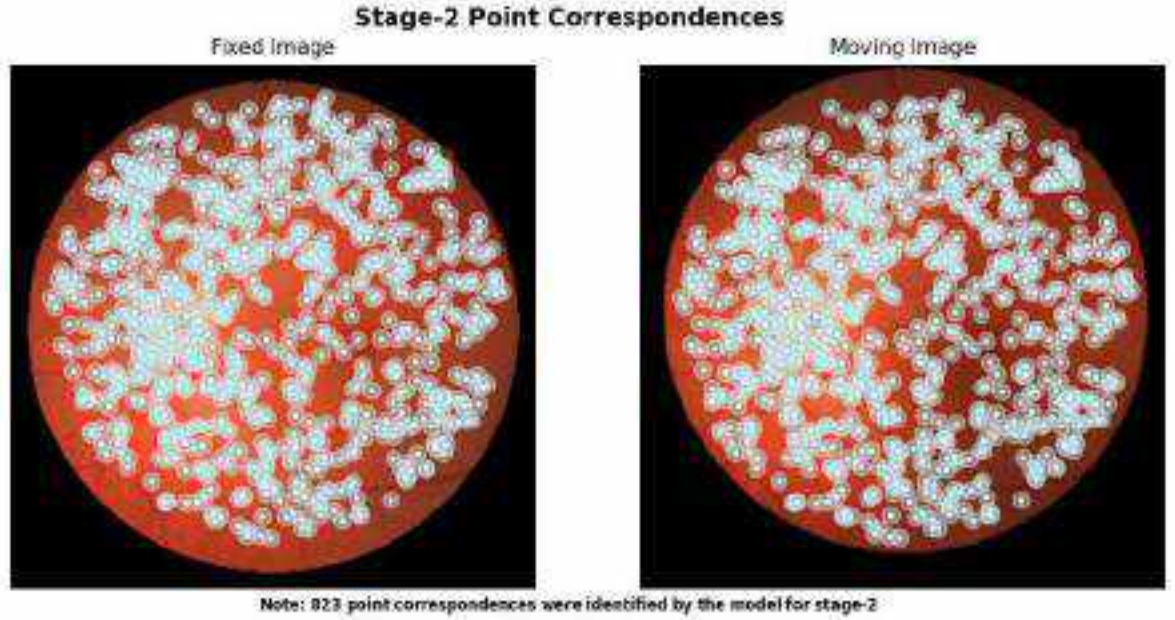

```
Mean Landmark Error for Case 61 Before Registration is 89.55993897021114 pixels
Mean Landmark Error for Case 61 After Registration is 2.0431388106410795 pixels
Case 62
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S63_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S63_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



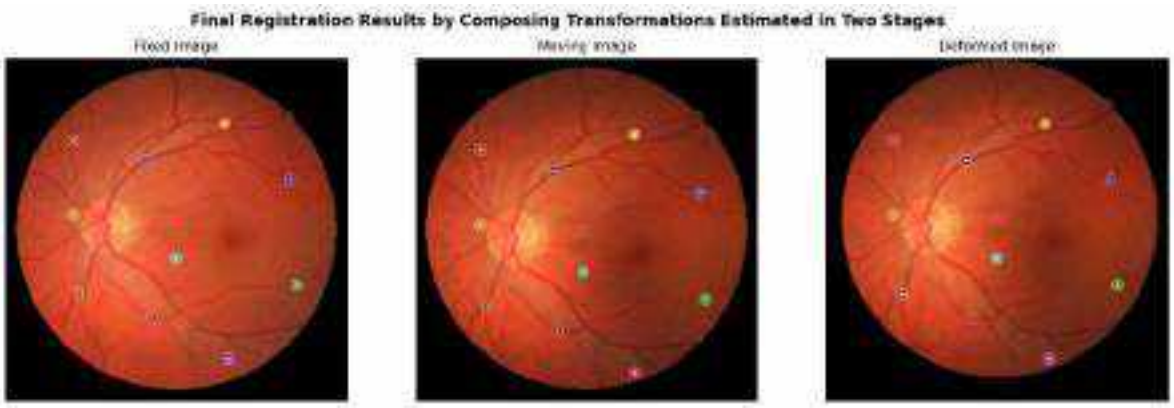
Note: 596 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.79537272e-01  3.96650952e-03  1.01068778e+01]
 [-1.37461491e-02  9.67407116e-01 -1.55084605e+01]
 [ 2.68762663e-06 -2.32748165e-05  1.00000000e+00]]
```

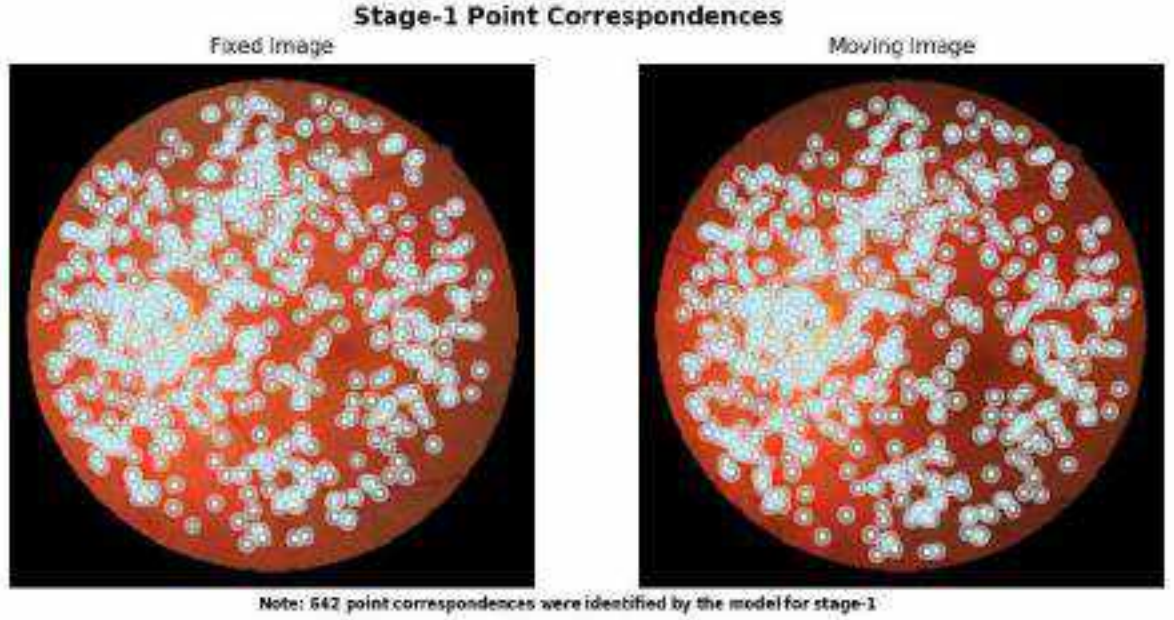

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



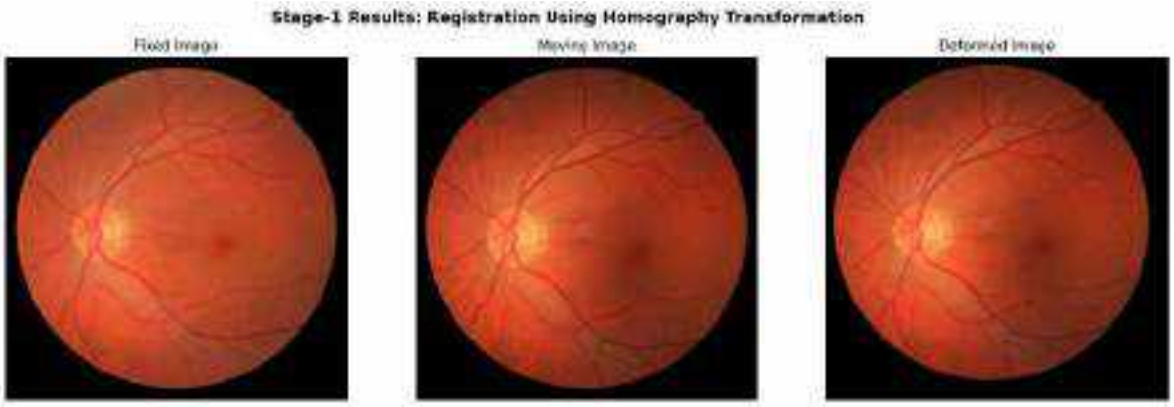
Note: 823 point correspondences were identified by the model for stage-2



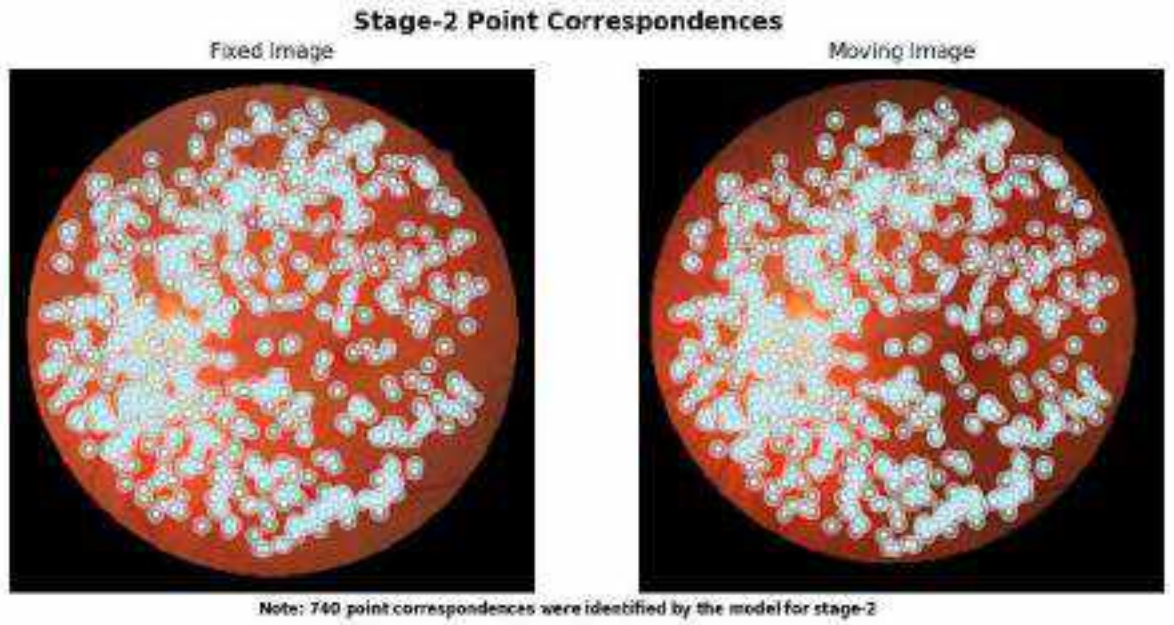

```
Mean Landmark Error for Case 62 Before Registration is 104.82242898931099 pixels
Mean Landmark Error for Case 62 After Registration is 2.0676687677164 63 pixels
Case 63
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S64_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S64_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



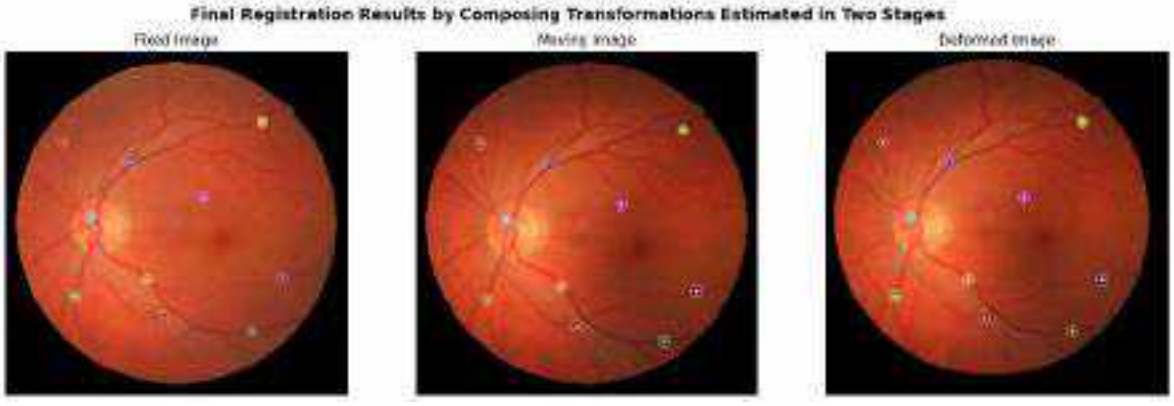

```
Homography Matrix:
[[ 9.76112683e-01  3.09177080e-02 -2.31372263e+01]
 [-4.01414570e-02  9.73292227e-01  9.89167431e+00]
 [-5.95323471e-06 -9.22744111e-06  1.00000000e+00]]
```



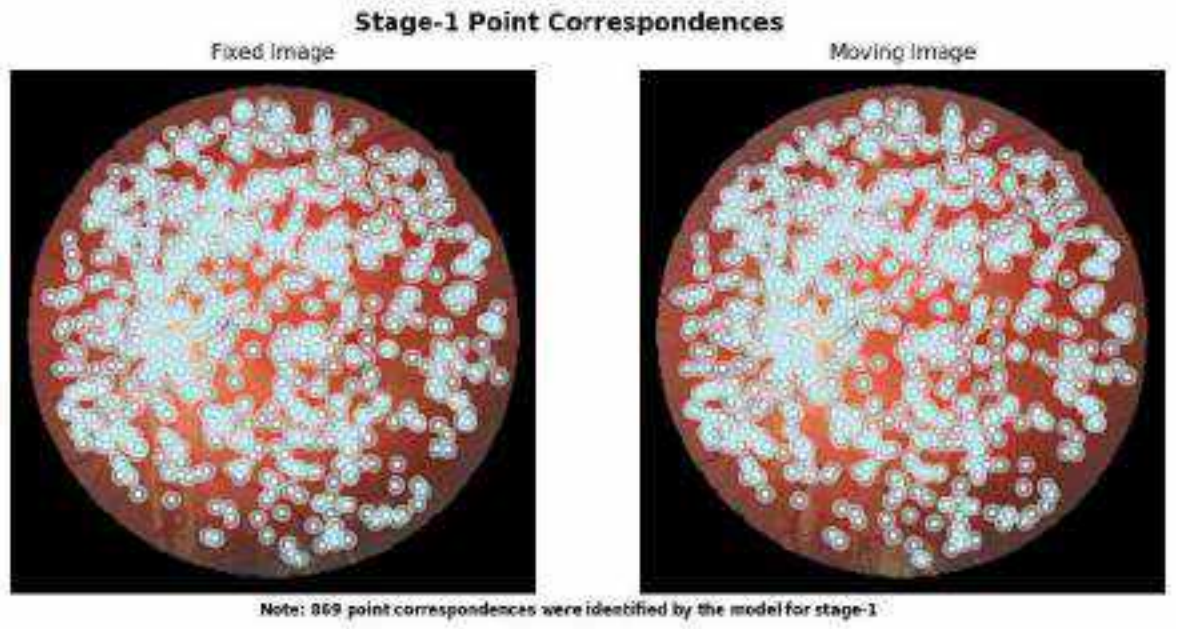

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



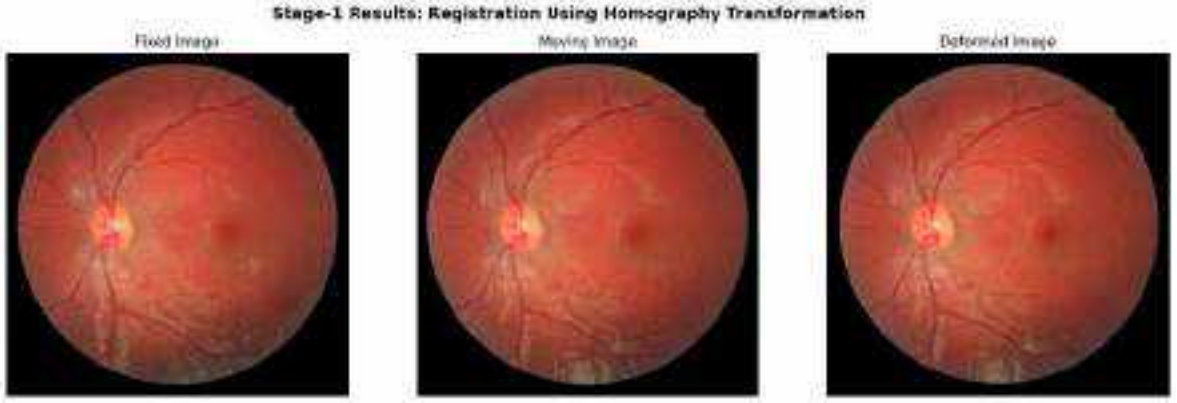
Note: 740 point correspondences were identified by the model for stage-2



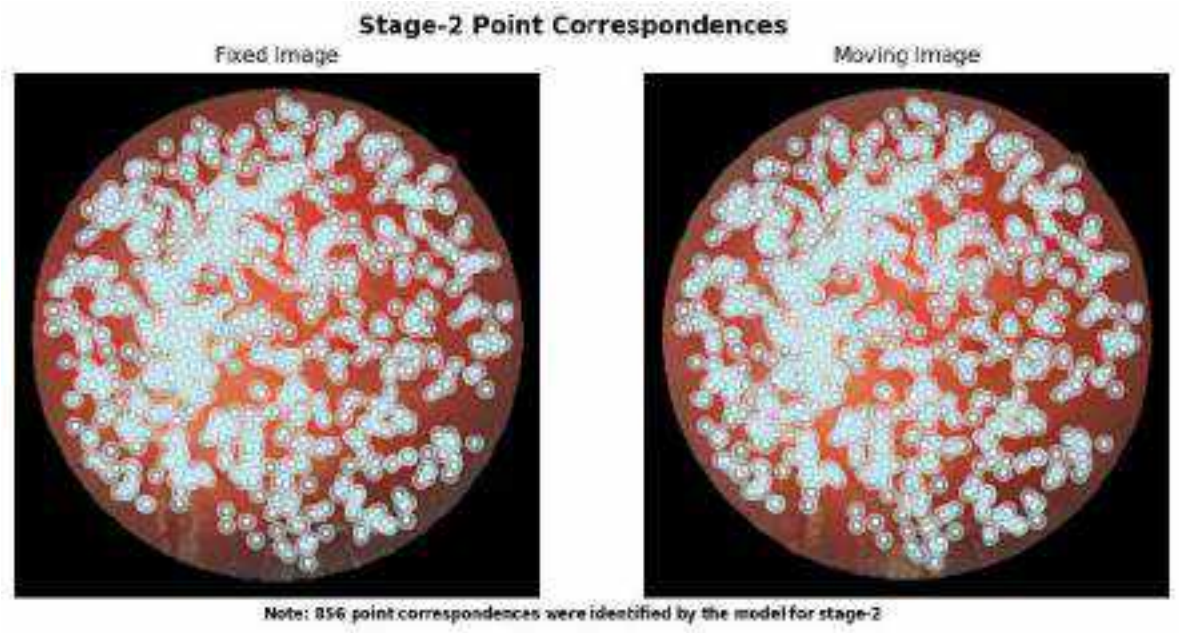
```
Mean Landmark Error for Case 63 Before Registration is 78.31674367970518 pixels
Mean Landmark Error for Case 63 After Registration is 1.4492396988704261 pixels
Case 64
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S65_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S65_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

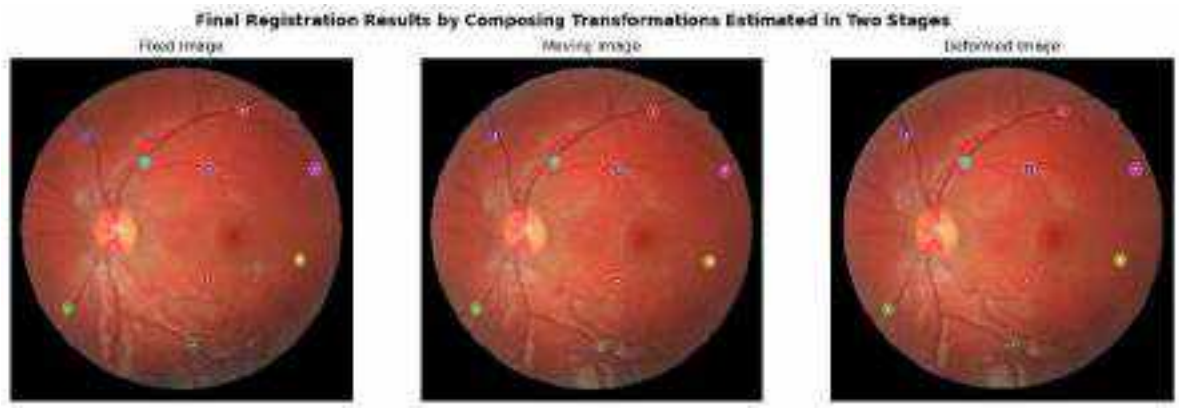



```
Homography Matrix:
[[ 1.00237685e+00  8.43125564e-03 -1.48924247e+00]
 [-4.11161933e-03  1.00028695e+00  1.34550484e+00]
 [ 4.65215902e-06  2.89447385e-07  1.00000000e+00]]
```

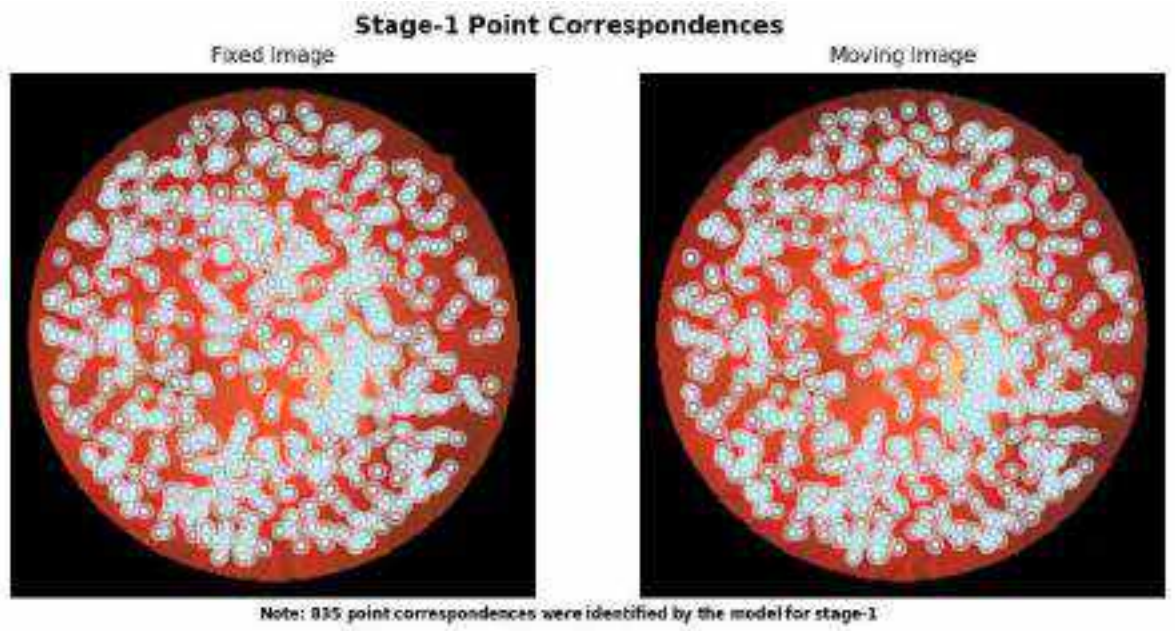



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```





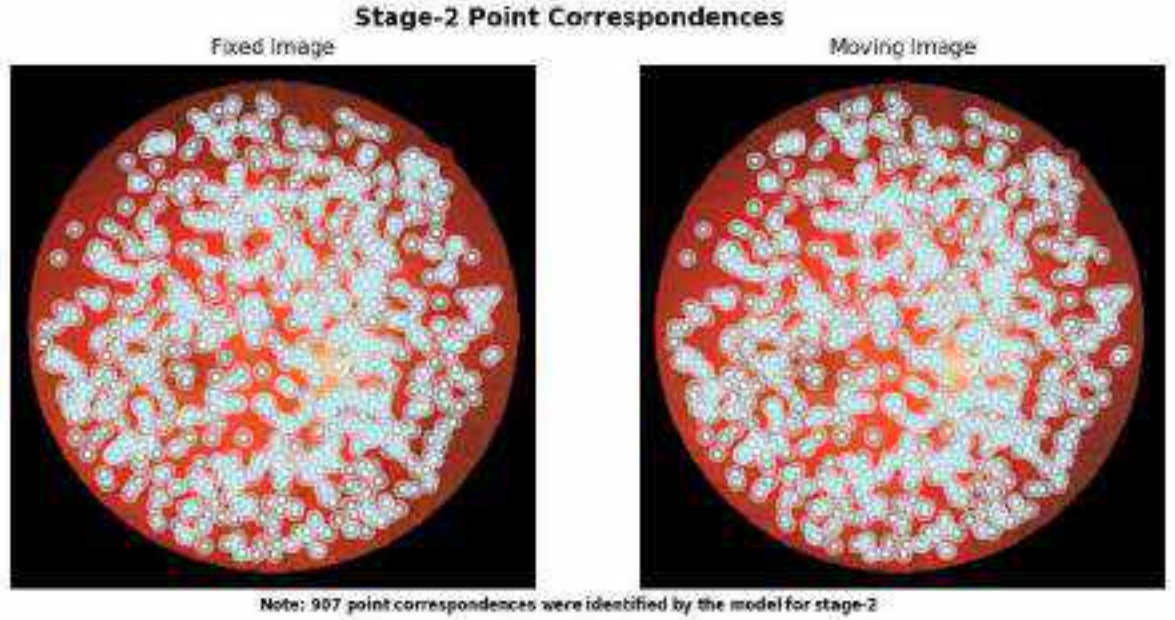

```
Mean Landmark Error for Case 64 Before Registration is 10.689249204188966 pixels
Mean Landmark Error for Case 64 After Registration is 1.7551884225162061 pixels
Case 65
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S66_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S66_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



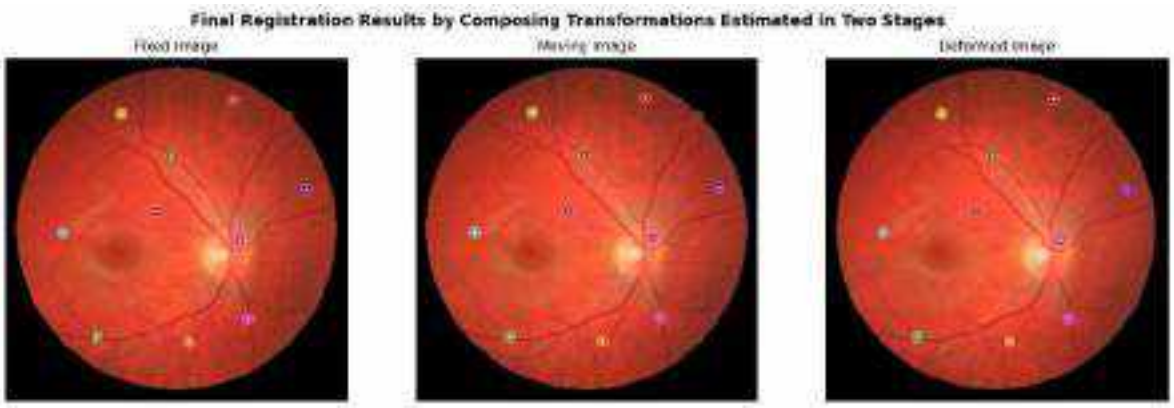
Note: 835 point correspondences were identified by the model for stage-1

```
Homography Matrix:
[[ 9.92252689e-01 -3.21175663e-03 -1.54382652e+00]
 [ 6.35169330e-04  9.96799318e-01  2.74321497e+00]
 [-6.50533639e-06  3.19296753e-06  1.00000000e+00]]
```

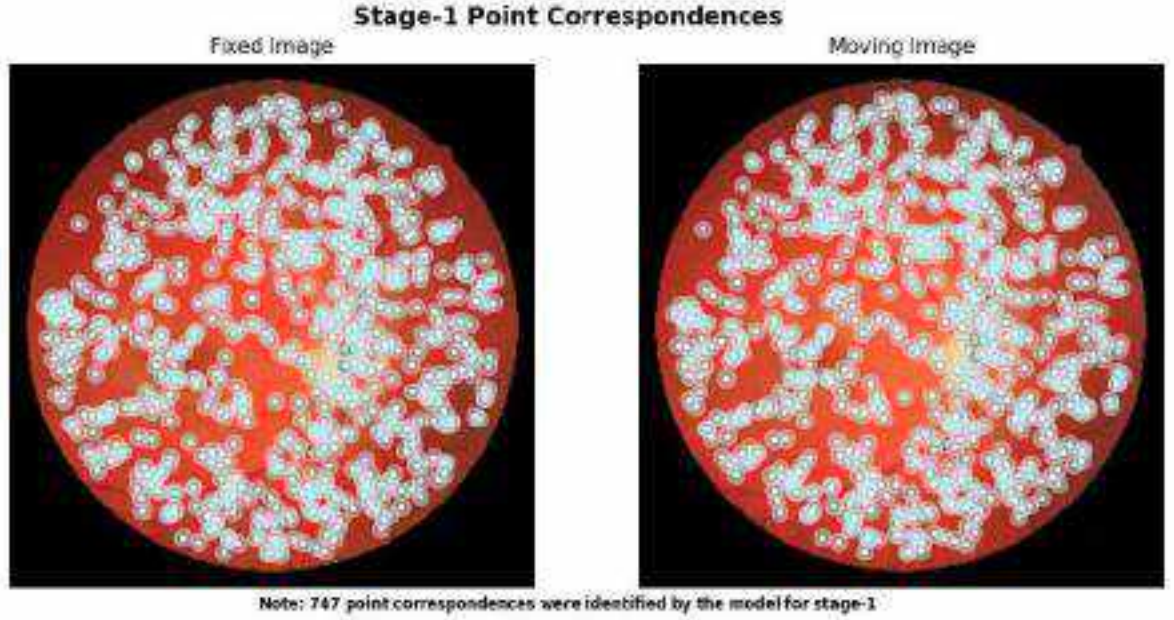

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



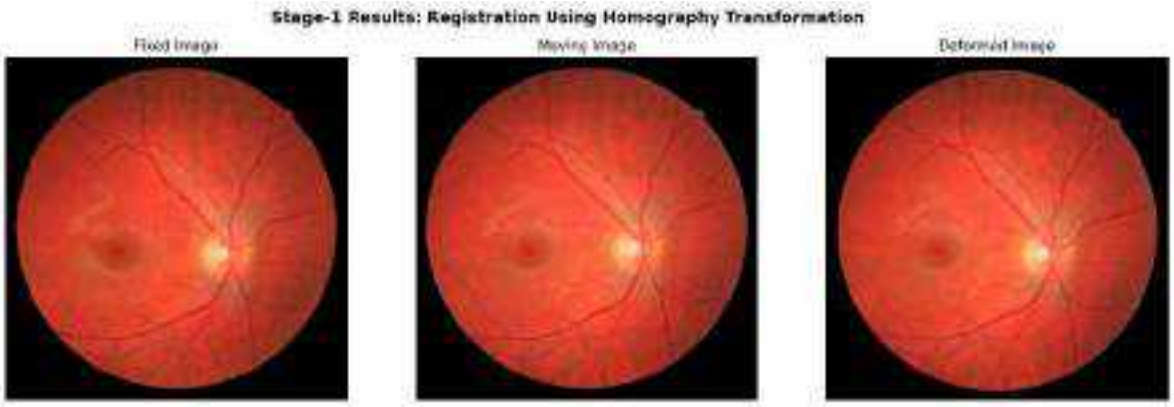
Note: 907 point correspondences were identified by the model for stage-2



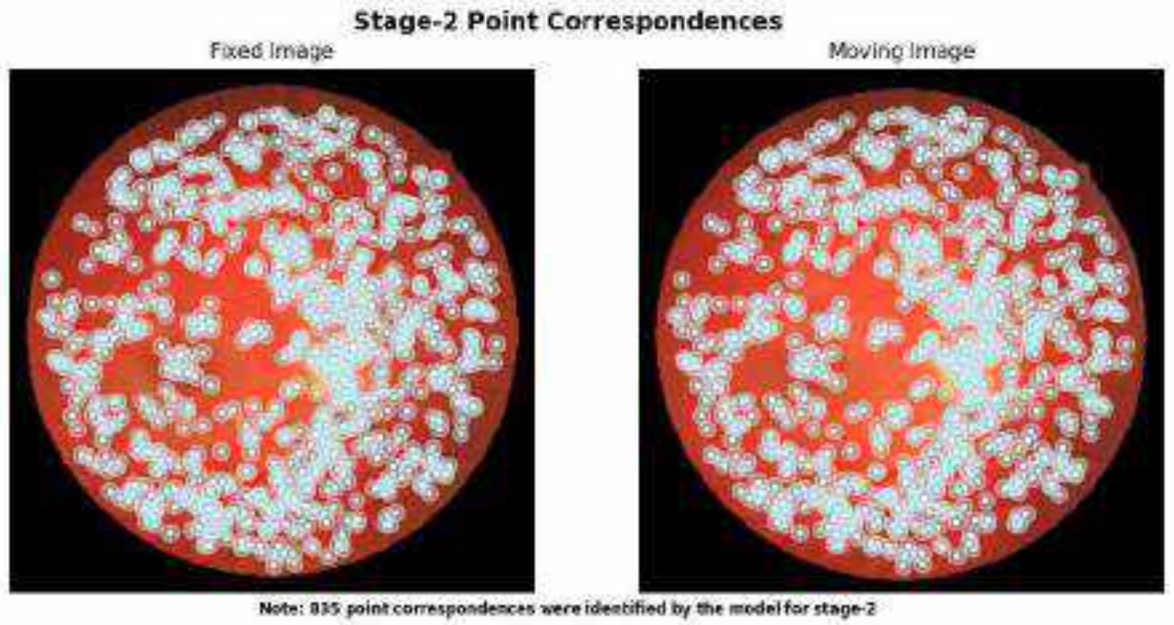

```
Mean Landmark Error for Case 65 Before Registration is 21.23051192025364 pixels
Mean Landmark Error for Case 65 After Registration is 1.9353003845732 36 pixels
Case 66
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S67_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S67_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



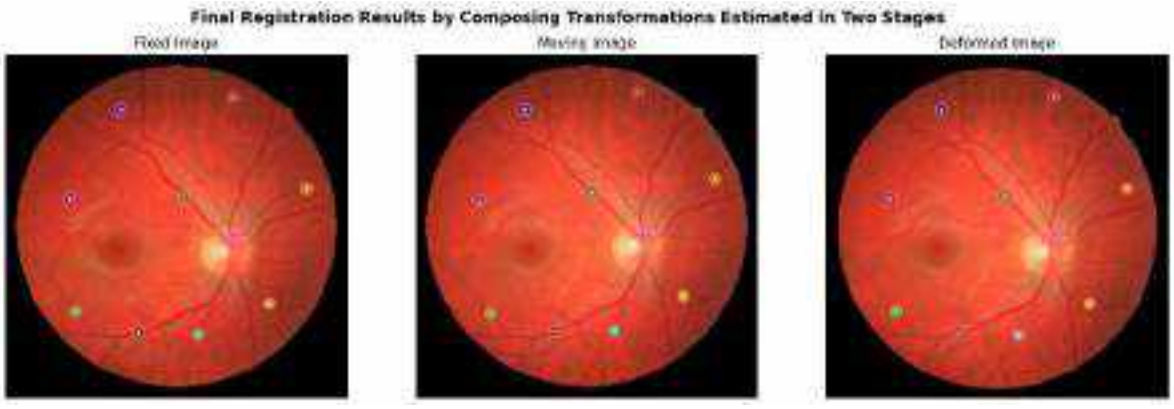

```
Homography Matrix:
[[ 9.96953807e-01 -4.65134265e-02  2.23395172e+01]
 [ 4.77997377e-02  1.00561631e+00 -1.51722287e+01]
 [-3.42250264e-06  8.75875726e-06  1.00000000e+00]]
```



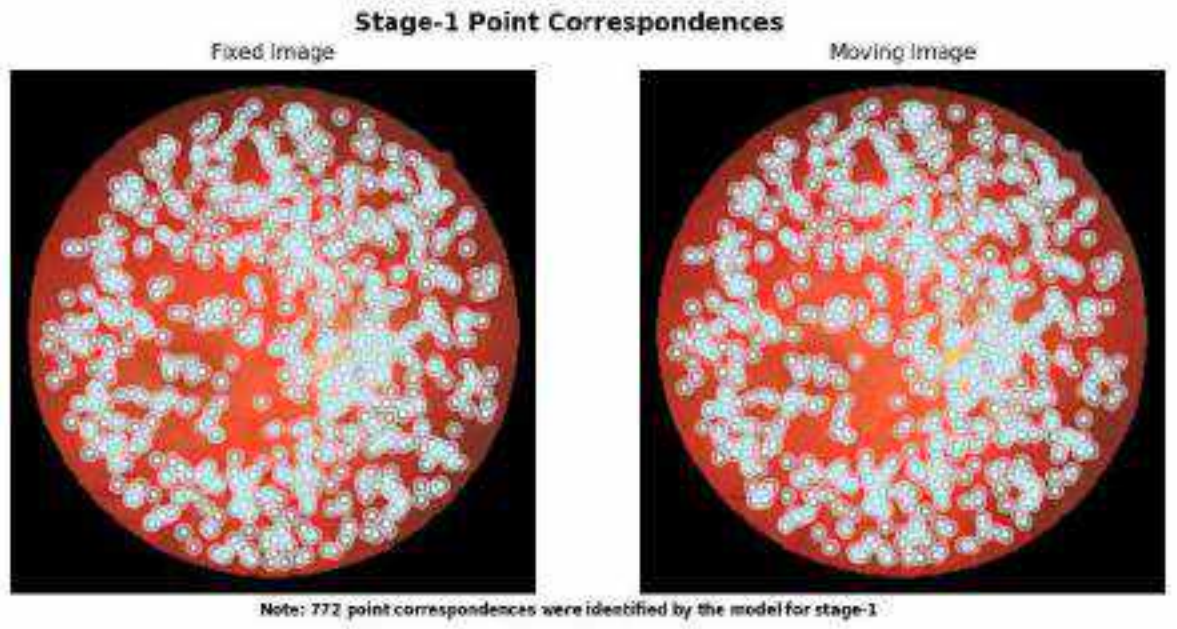

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



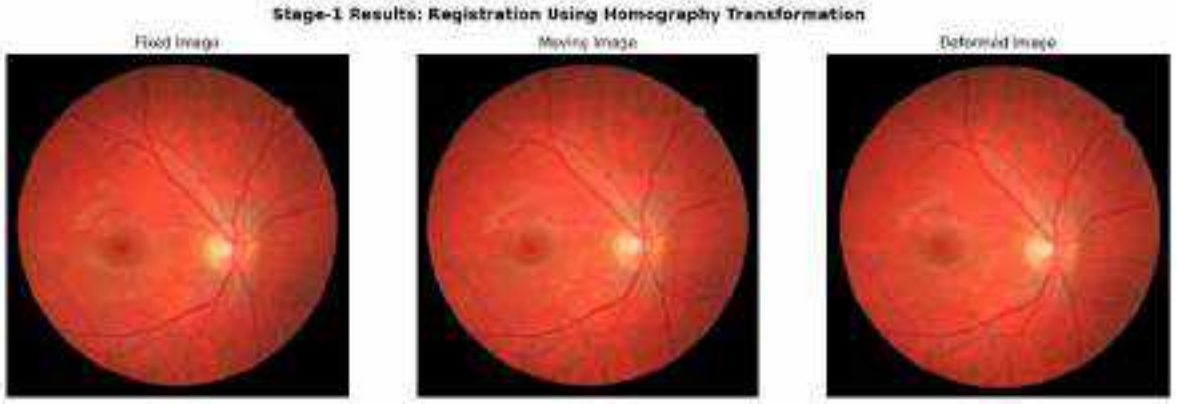
Note: 835 point correspondences were identified by the model for stage-2



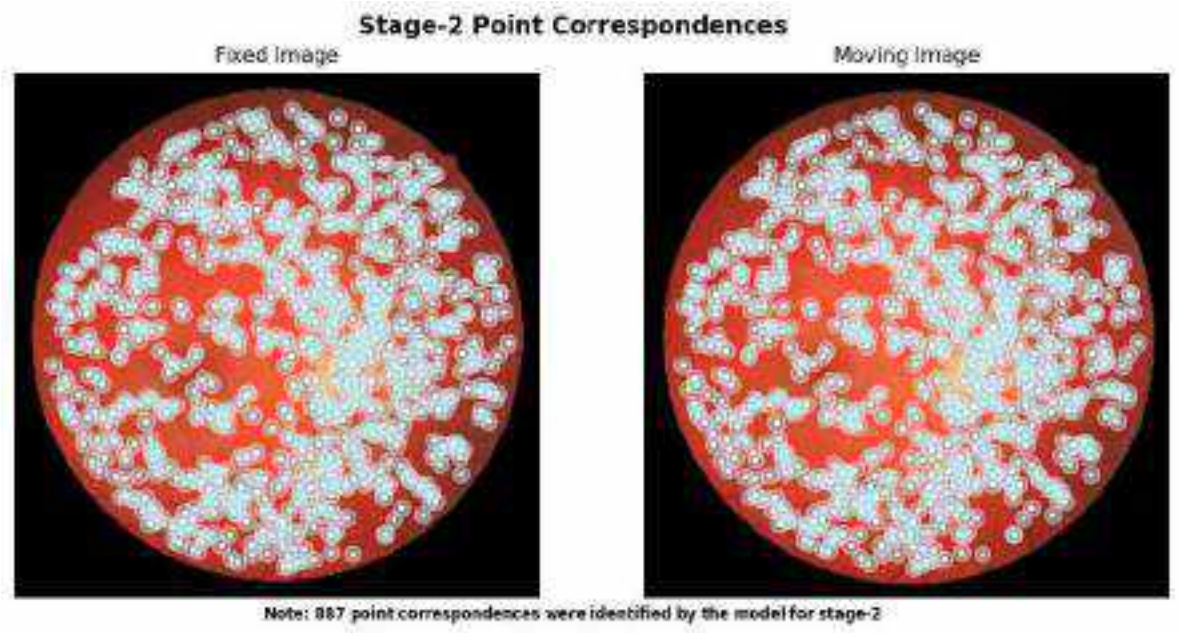
```
Mean Landmark Error for Case 66 Before Registration is 52.254740688939682 pixels
Mean Landmark Error for Case 66 After Registration is 1.5851936423545667 pixels
Case 67
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S68_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S68_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

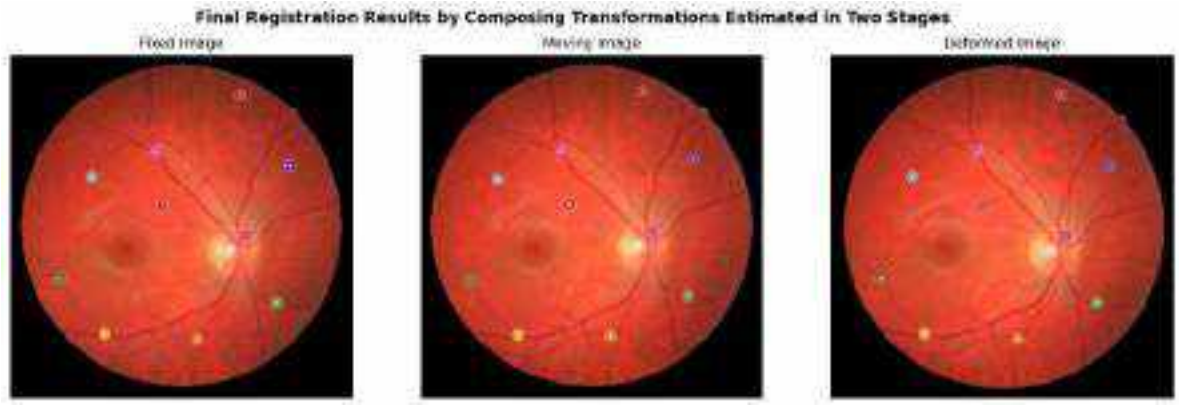



```
Homography Matrix:
[[ 1.00596776e+00 -4.24633630e-02  2.40760074e+01]
 [ 4.67901051e-02  1.00785071e+00 -1.79217561e+01]
 [ 2.85184423e-06  4.82698516e-06  1.00000000e+00]]
```

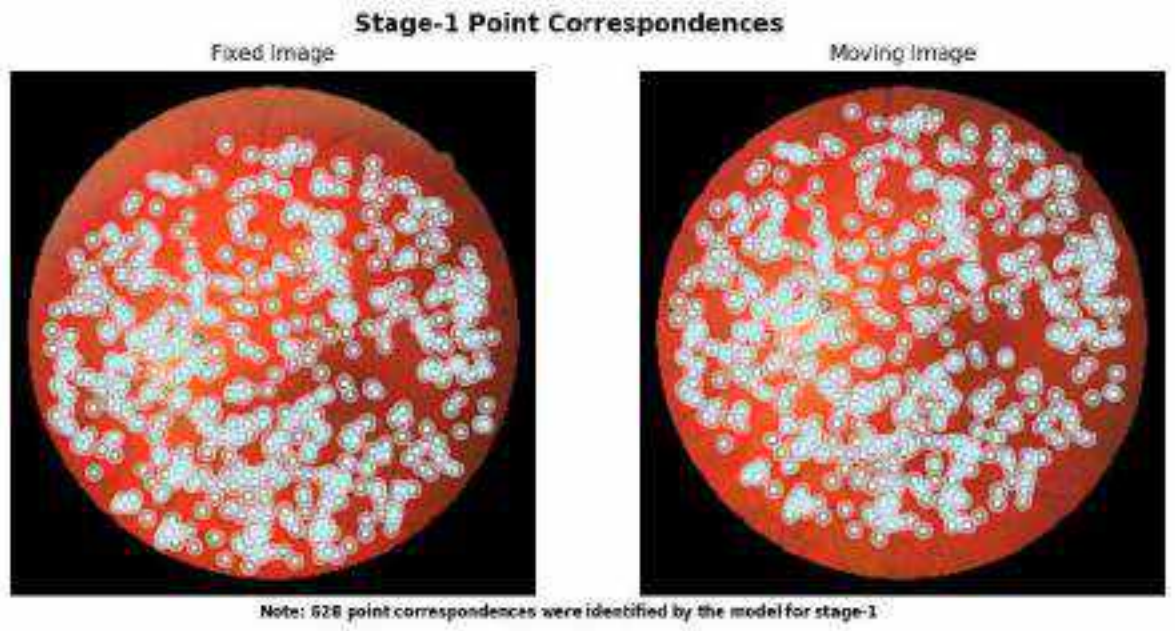



```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



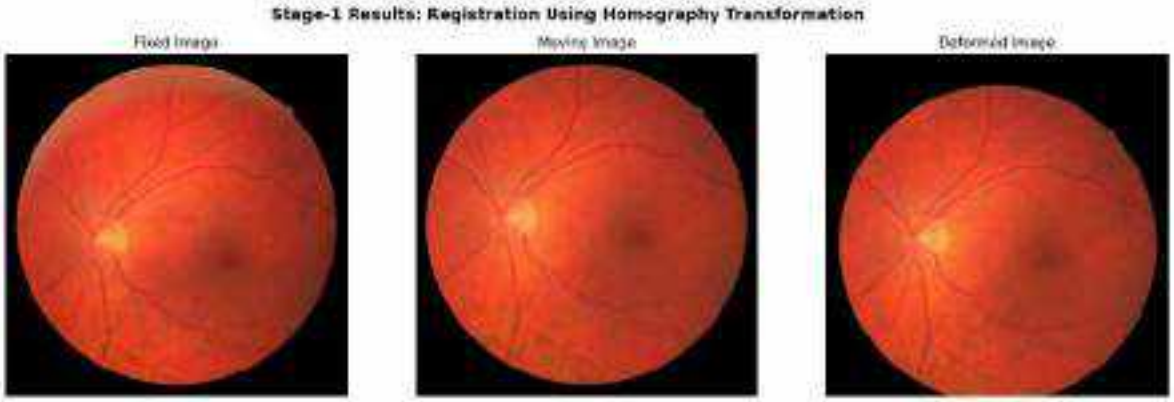



```
Mean Landmark Error for Case 67 Before Registration is 45.455242939583144 pixels
Mean Landmark Error for Case 67 After Registration is 1.3000938430789204 pixels
Case 68
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S69_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S69_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

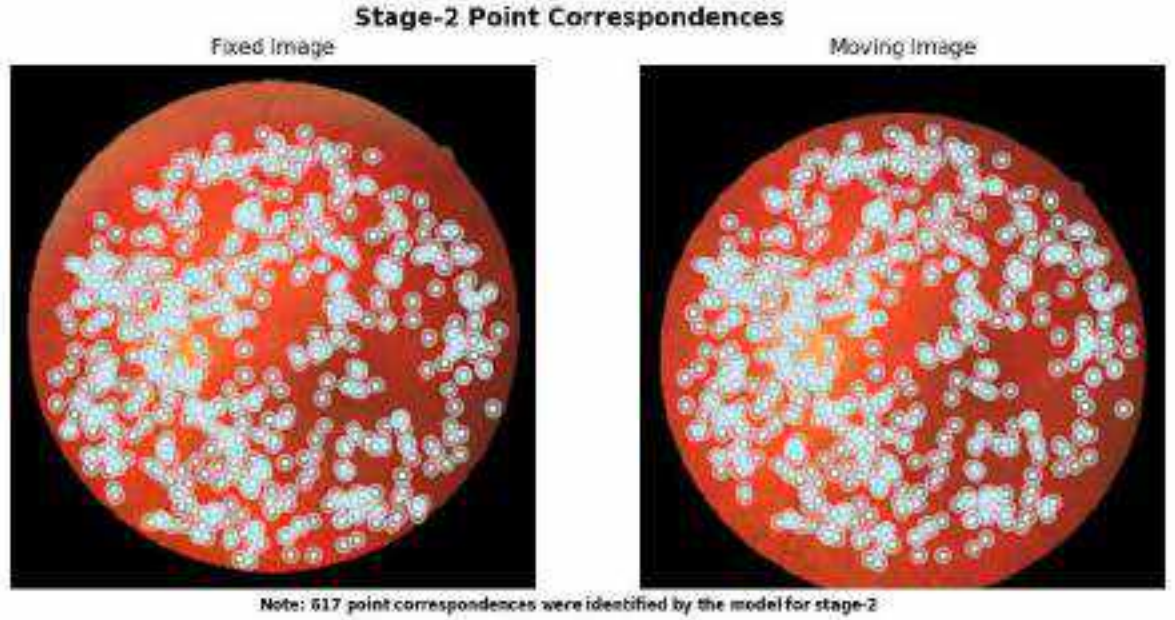


Note: 526 point correspondences were identified by the model for stage-1

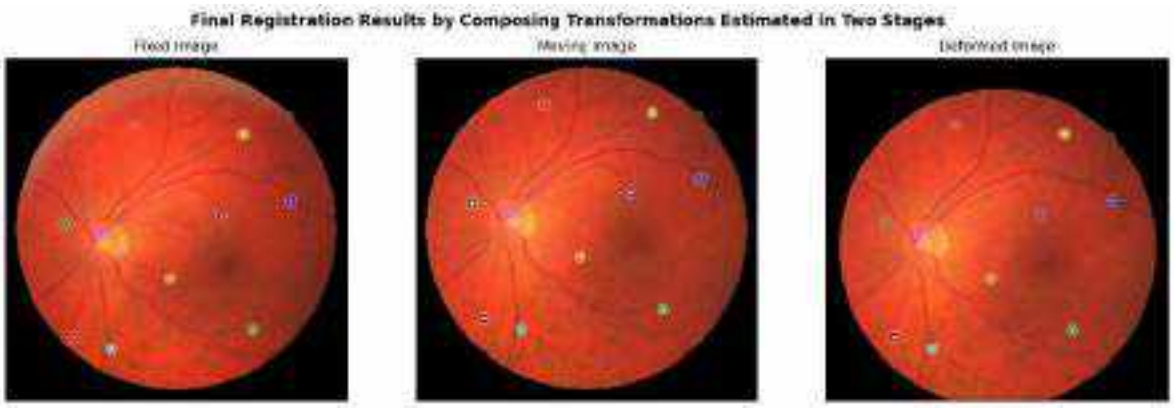
```
Homography Matrix:
[[1.00243631e+00 7.09223185e-04 4.81525259e+00]
 [9.82698120e-03 1.01323718e+00 5.11767236e+01]
 [1.39703216e-07 2.09590976e-05 1.00000000e+00]]
```



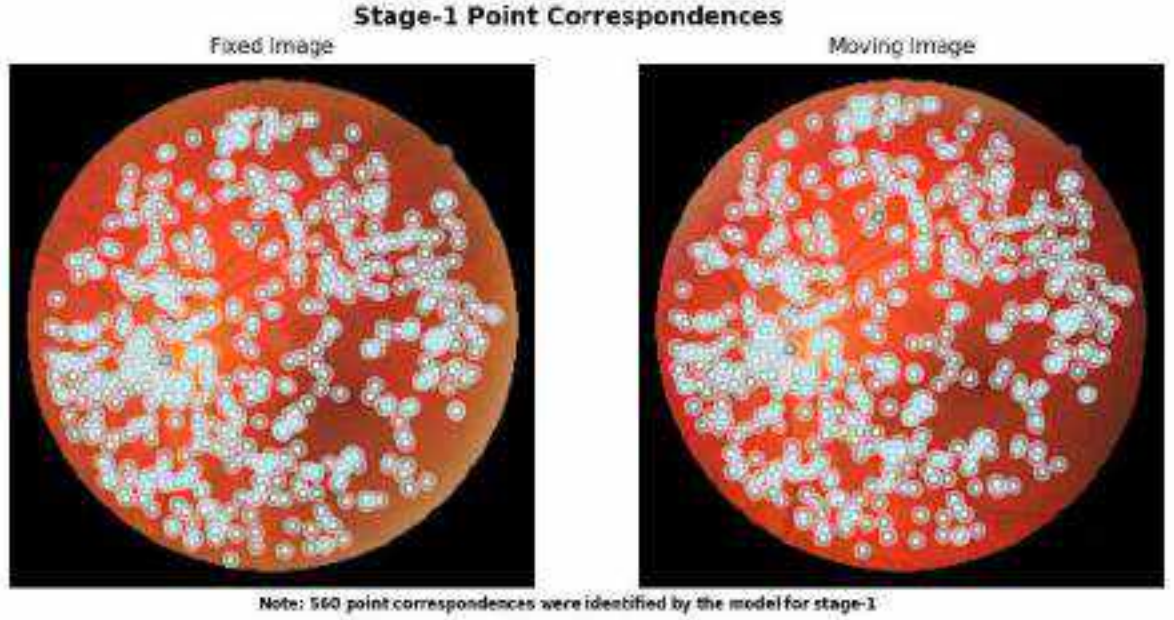

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 617 point correspondences were identified by the model for stage-2

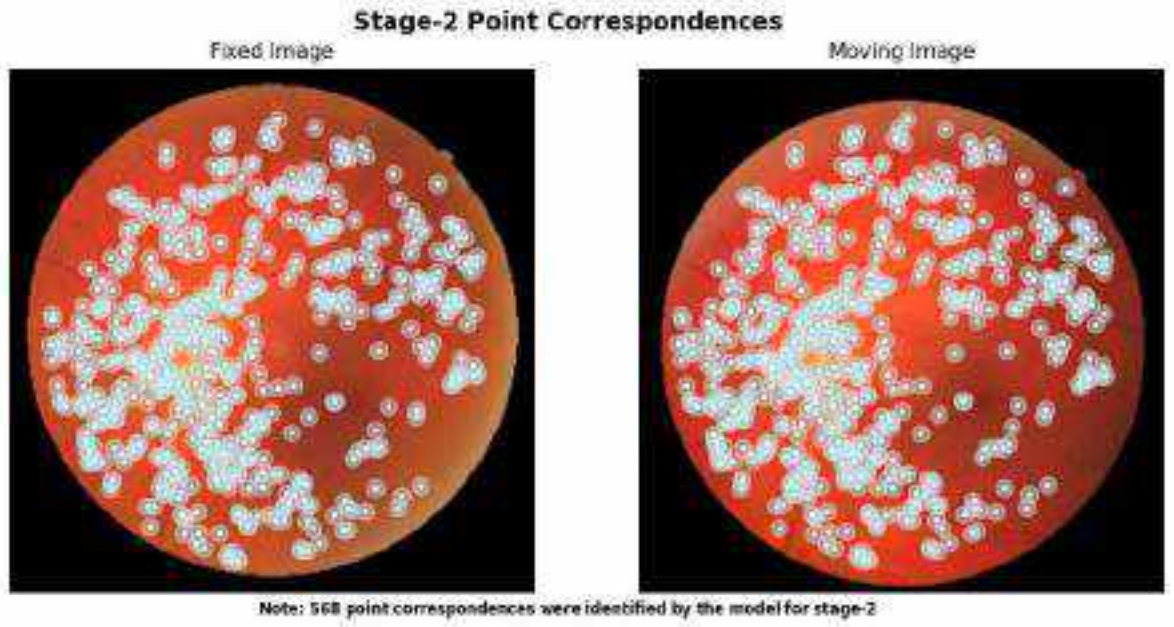



```
Mean Landmark Error for Case 68 Before Registration is 176.87273984837665 pixels
Mean Landmark Error for Case 68 After Registration is 2.2583032979797086 pixels
Case 69
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S70_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S70_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```

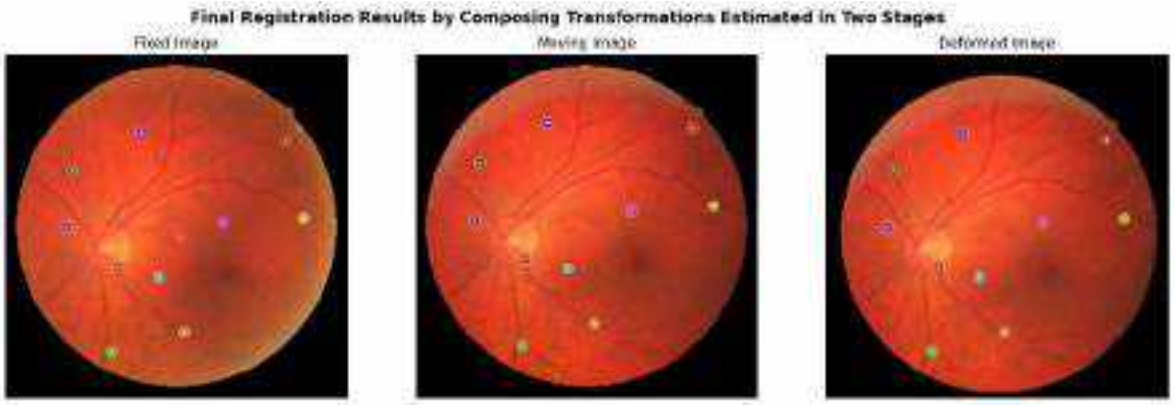



```
Homography Matrix:
[[ 1.00049243e+00 -1.45344367e-02  1.65341317e+01]
 [ 2.64427514e-02  1.00739811e+00  1.58911150e+01]
 [ 5.35024082e-06  1.96380050e-05  1.00000000e+00]]
```

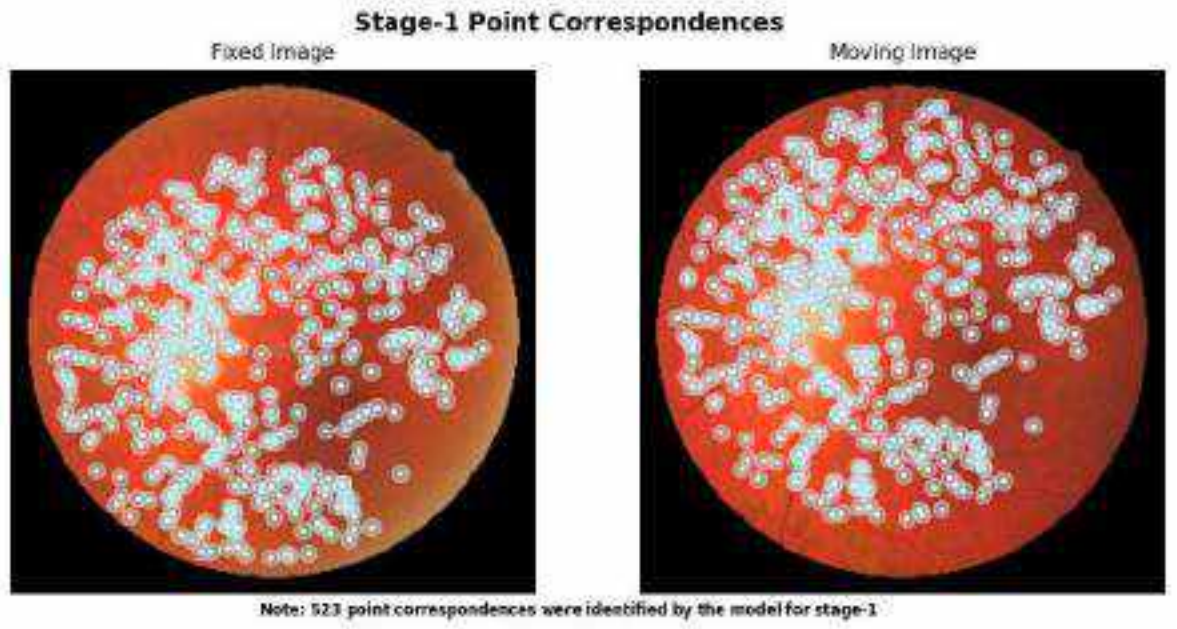

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



Note: 568 point correspondences were identified by the model for stage-2

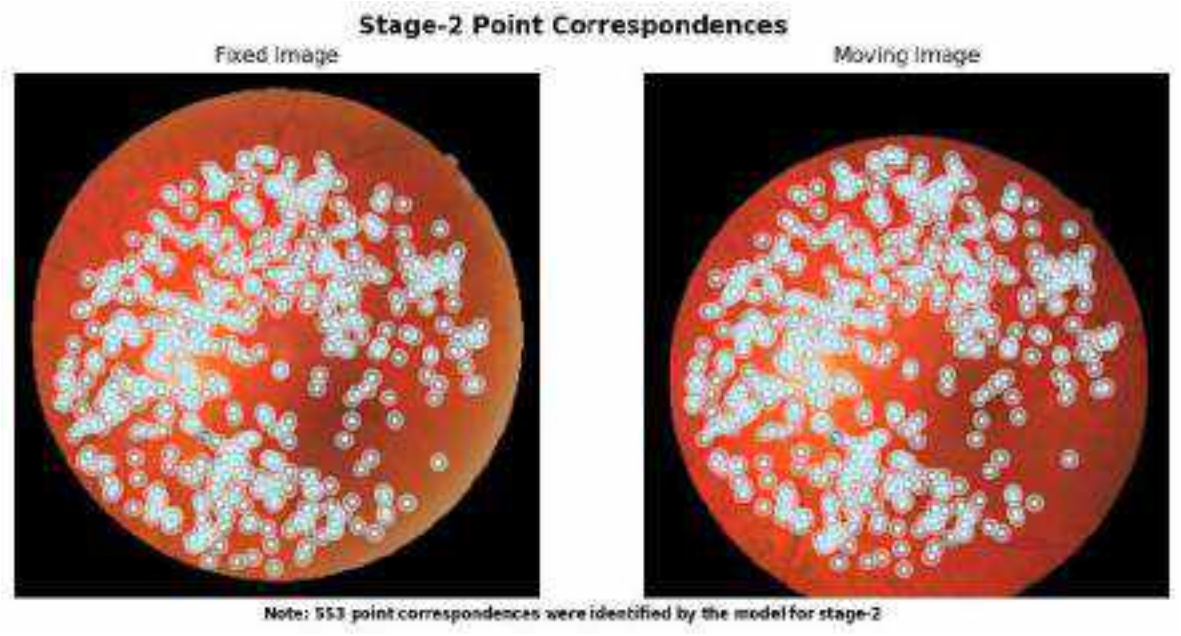


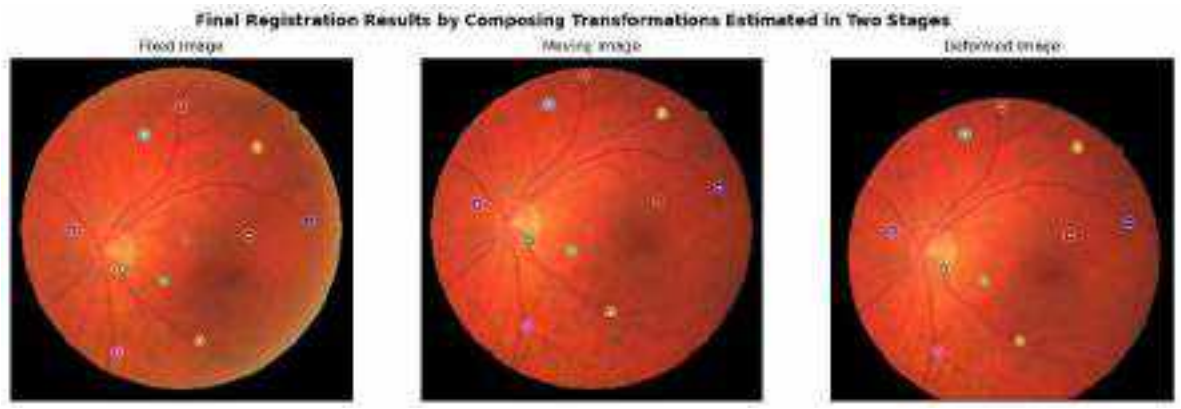
```
Mean Landmark Error for Case 69 Before Registration is 80.72650659506903 pixels
Mean Landmark Error for Case 69 After Registration is 2.39143301098774 pixels
Case 70
Loading Fixed Images /blue/weishao/vi.sivaraman/Tasks/Image_Registration/2D-Regis
tration/RetinaRegNet/Fire_Registration/FIRE/Images/S71_1.jpg Moving Image/blue/we
ishao/vi.sivaraman/Tasks/Image_Registration/2D-Registration/RetinaRegNet/Fire_Reg
istration/FIRE/Images/S71_2.jpg to the framework
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```




```
Homography Matrix:
[[ 1.00054506e+00 -1.24464403e-02  2.19311062e+01]
 [ 3.63015794e-02  1.02773349e+00  6.57441024e+01]
 [ 4.49707124e-06  4.59802329e-05  1.00000000e+00]]
```

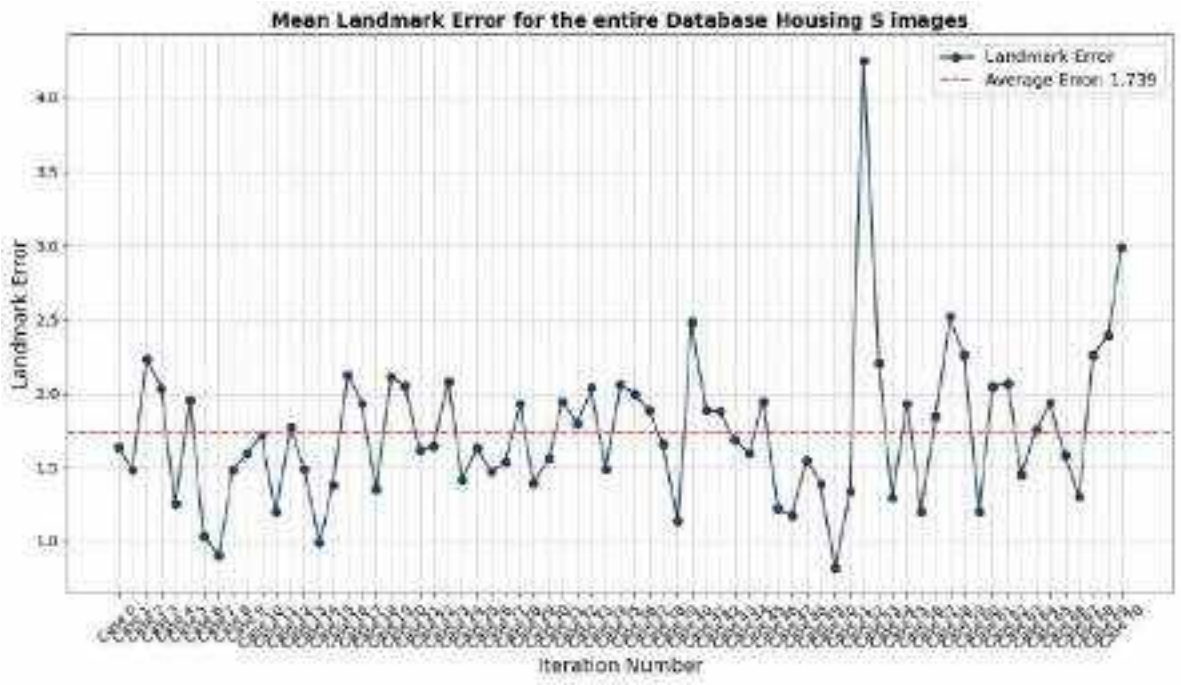

```
Loading pipeline components...:   0%|          | 0/6 [00:00<?, ?it/s]
```



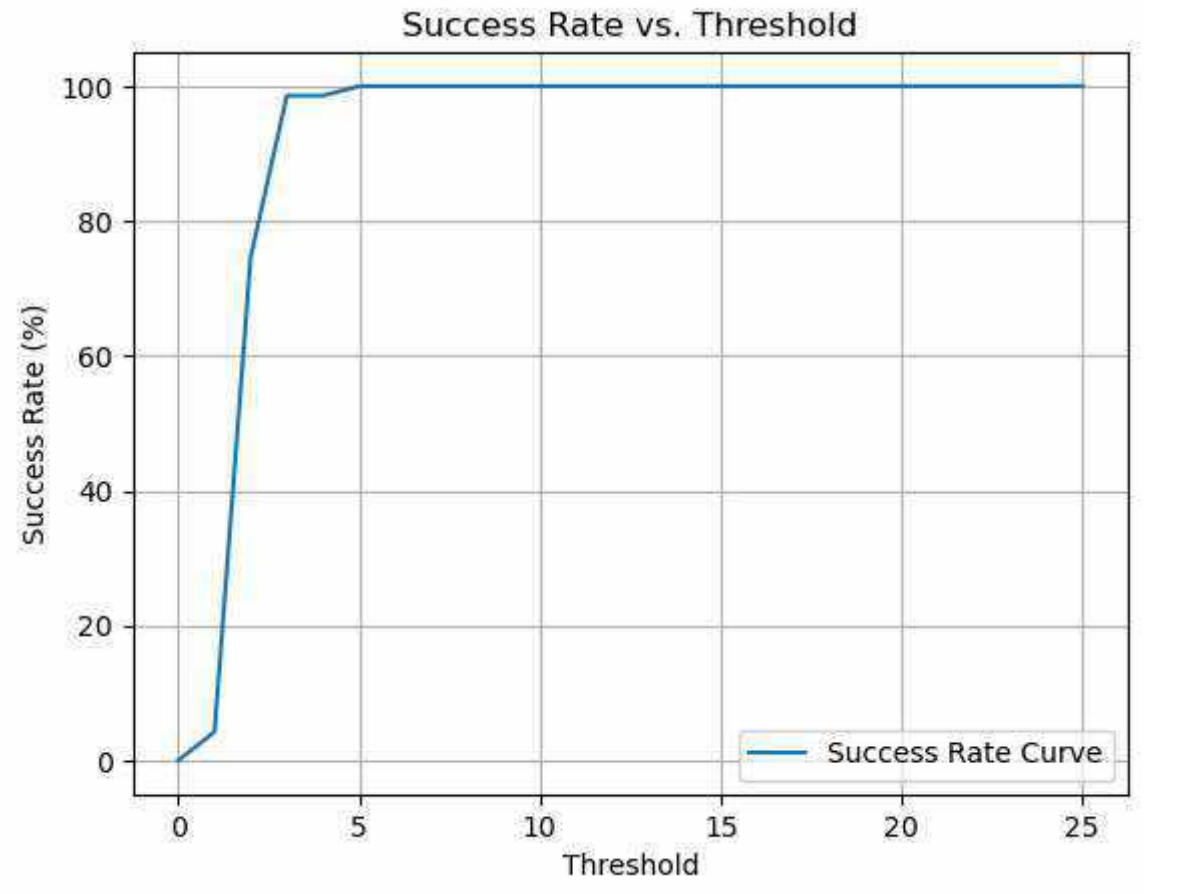



```
Mean Landmark Error for Case 70 Before Registration is 259.56947058950755 pixels
Mean Landmark Error for Case 70 After Registration is 2.98961063777418 pixels
```

In [26]:
```
plot_landmark_errors(landmark_errors3,os.path.join(os.getcwd(),'FIRE_Image_Regis
```

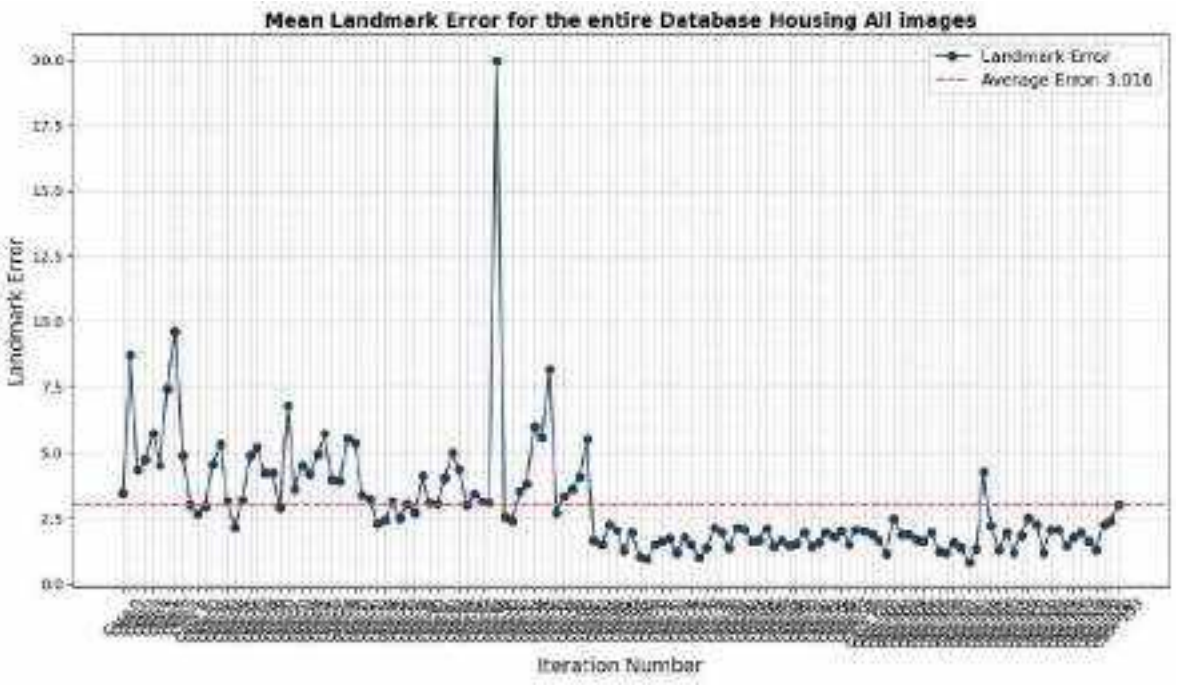


In [27]:
```
compute_plot_FIRE_AUC(landmark_errors3)
```

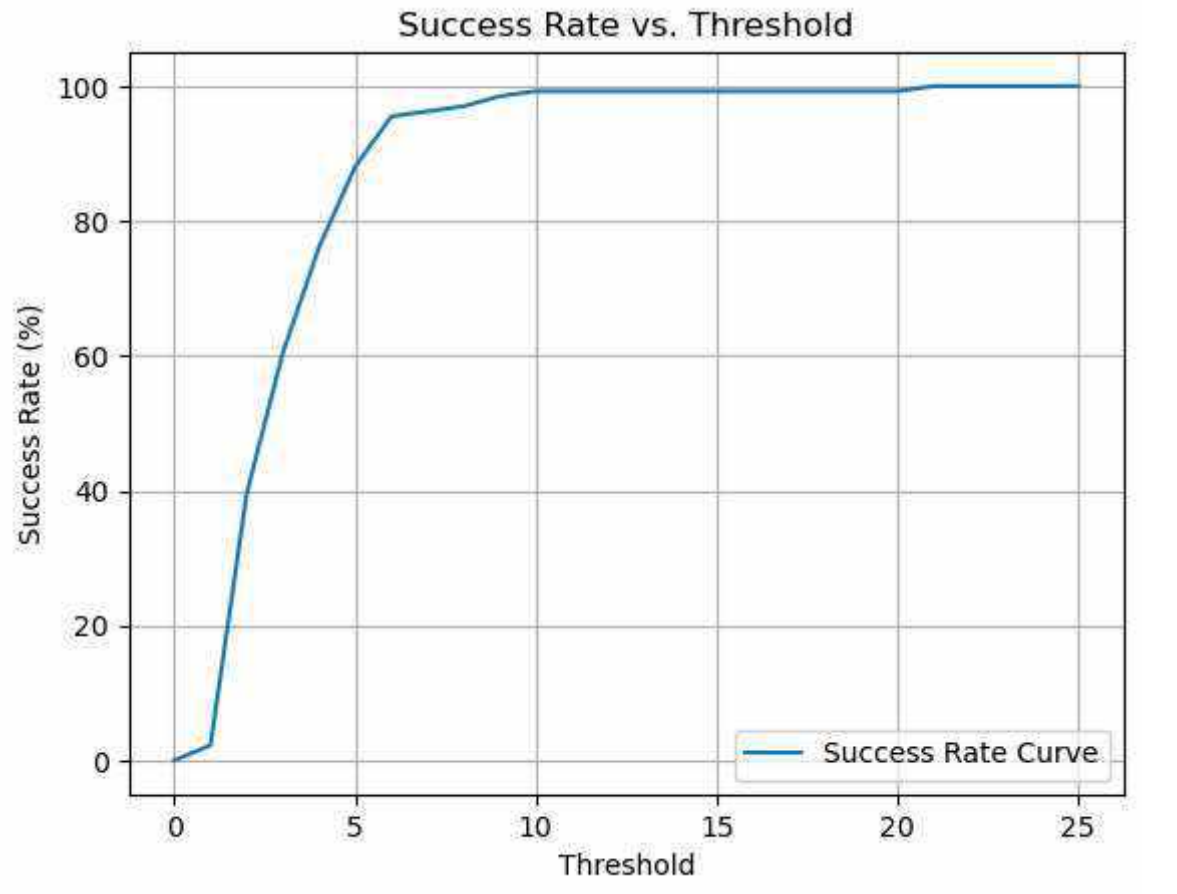



```
AUC: 0.9504225352112676
```

In [28]:
```
landmark_errors = landmark_errors1 + landmark_errors2 + landmark_errors3
plot_landmark_errors(landmark_errors,os.path.join(os.getcwd(),'FIRE_Image_Regist
```



In [29]:
```
compute_plot_FIRE_AUC(landmark_errors)
```




```
AUC: 0.8982089552238806
```