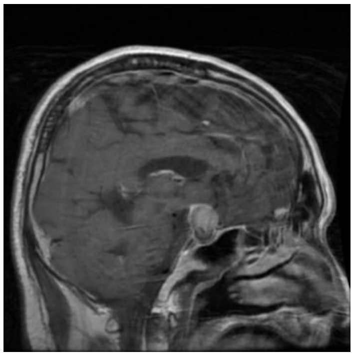
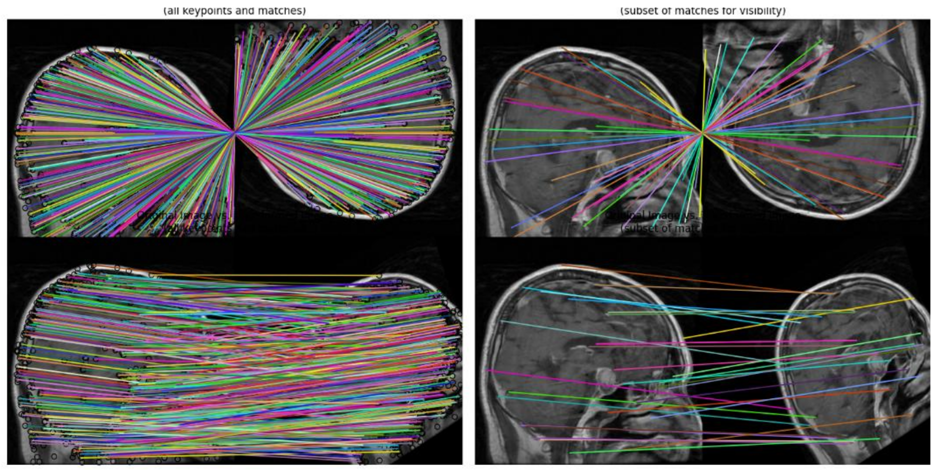
Caden Thompson HW 4 4/29/25

Image 1:

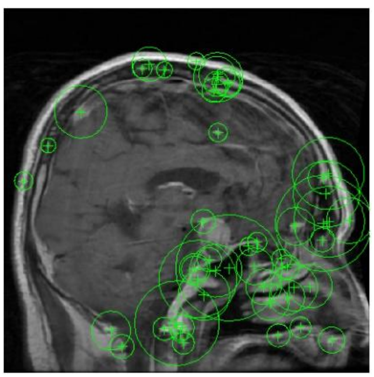
Original



SIFT



SURF



ORB

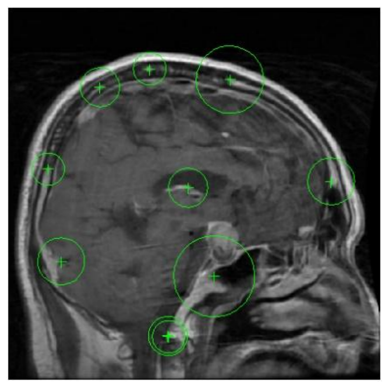
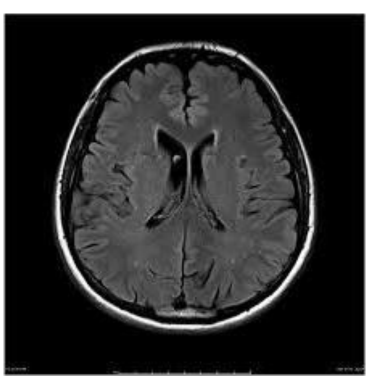
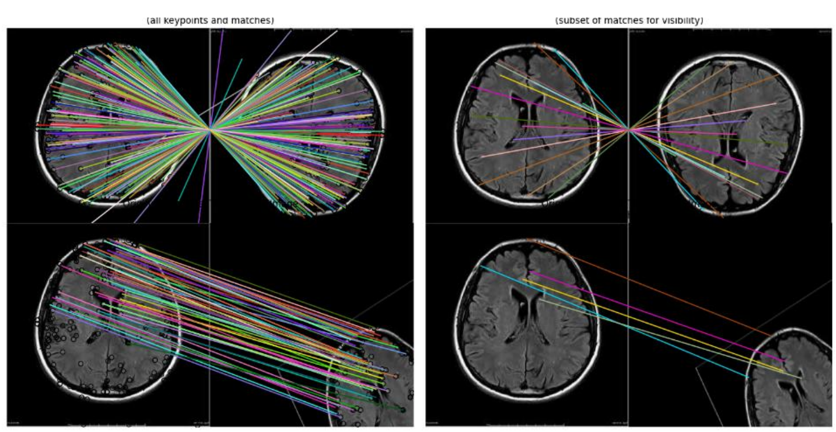


Image 2:

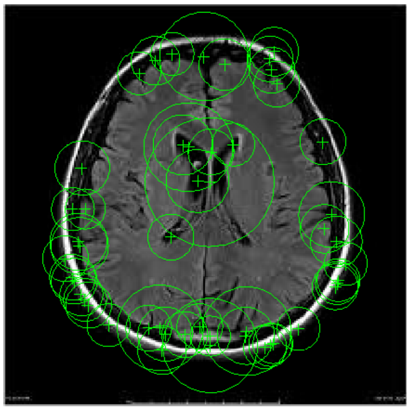
Original



SIFT



SURF



ORB

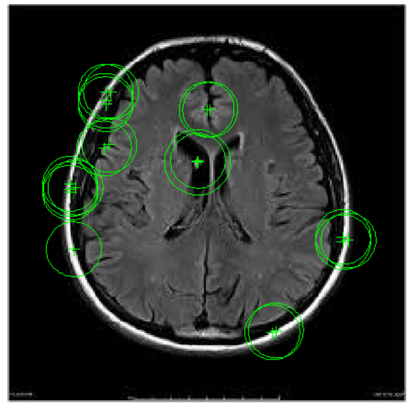
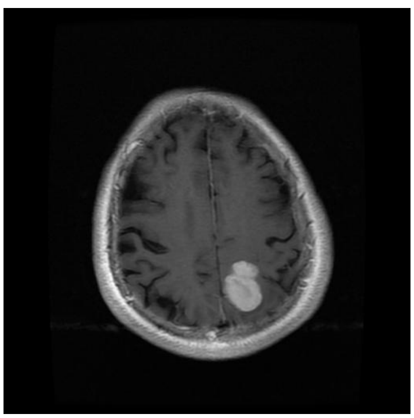
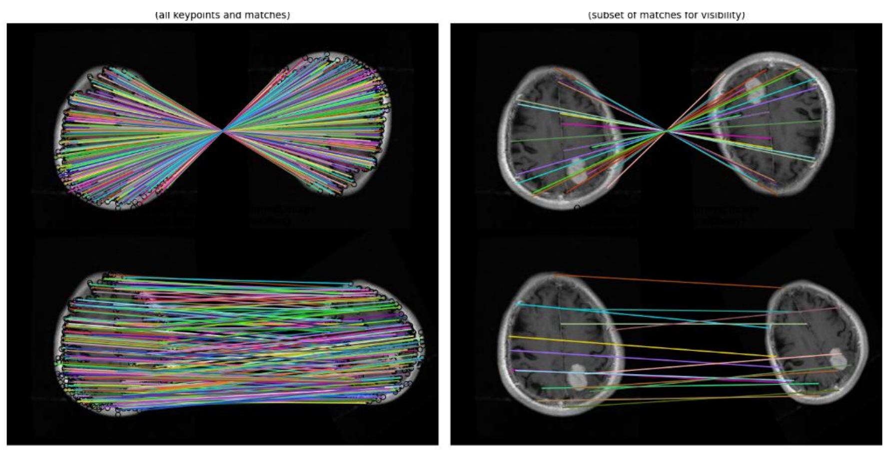


Image 3:

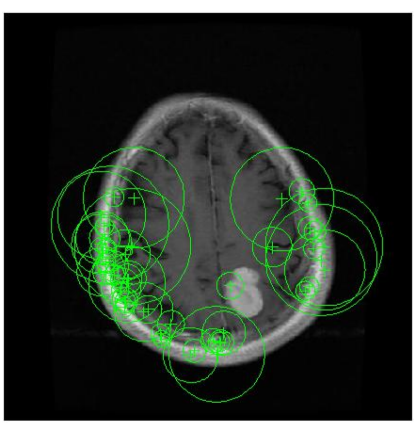
Original



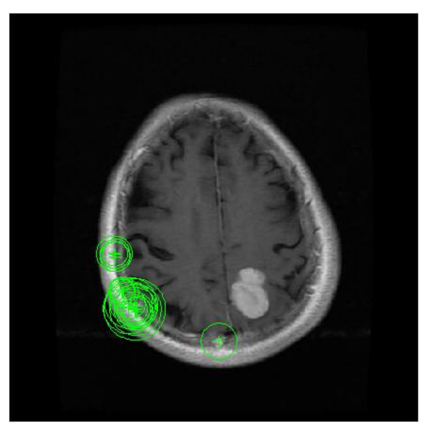
SIFT



SURF



ORB



Analysis:

Image1:

For image 1 all three methods were able to find key points. However, the best key point detection was produced by SIFT. SIFT seemed to produce the best overall key points. The key points were well distributed, and it finds small details well. It also did not miss any important areas. The SURF method seemed to do well in some areas, but it missed many smaller key points in the brain area especially the center. The ORB method produced the least number of key points, and it seemed to miss any key points in the nasal area of the scan.

Image 2:

For image 2 SIFT also seemed to produce the best key points. In this image however, a close second was SURF, the results were well distributed and seemed to capture most of the key points in the image. Orb performed very poorly on this image, resulting in key points that were not reflected well on both sides, with most of the key points being on the left side. Additionally, there were just few key points, this would result in rather poor performance in the case that the image is rotated or modified in a significant way.

Image 3:

Again, SIFT has the most even distribution of key points and finds smaller details better than the other two methods. SURF again performed well, just not as well as SIFT. The key points are sufficient, and it also found the detail in the brain which is important. The ORB method performed poorly once more. It did not find the key point in the brain, and all of the key points again are on one side of the image rather than distributed. Having all of the key points in one area of the image is not ideal.

Caden Thompson HW 4 4/29/25

Implementation:

SIFT

```python
import matplotlib.pyplot as plt
import cv2
from skimage import data
from skimage import transform
from skimage.color import rgb2gray
from skimage.feature import match_descriptors, plot_matched_features, SIFT
#https://scikit-image.org/docs/0.25.x/auto_examples/features_detection/plot_sift.html
imagepath = 'C:/Users/caden/Downloads/archive/Training/meningioma/Tr-me_1286.jpg'
img1 = cv2.imread(imagepath, 0)
img2 = transform.rotate(img1, 180)
tform = transform.AffineTransform(scale=(1.3, 1.1), rotation=0.5, translation=(0, -200))
img3 = transform.warp(img1, tform)
descriptor_extractor = SIFT()
descriptor_extractor.detect_and_extract(img1)
keypoints1 = descriptor_extractor.keypoints
descriptors1 = descriptor_extractor.descriptors
descriptor_extractor.detect_and_extract(img2)
keypoints2 = descriptor_extractor.keypoints
descriptors2 = descriptor_extractor.descriptors
descriptor_extractor.detect_and_extract(img3)
keypoints3 = descriptor_extractor.keypoints
descriptors3 = descriptor_extractor.descriptors
matches12 = match_descriptors(
    descriptors1, descriptors2, max_ratio=0.6, cross_check=True
)
matches13 = match_descriptors(
    descriptors1, descriptors3, max_ratio=0.6, cross_check=True
)
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(11, 8))
plt.gray()
plot_matched_features(
    img1,
    img2,
    keypoints0=keypoints1,
    keypoints1=keypoints2,
    matches=matches12,
    ax=ax[0, 0],
)
ax[0, 0].axis('off')
ax[0, 0].set_title("Original Image vs. Flipped Image\n" "(all keypoints and matches)")
plot_matched_features(
    img1,
    img3,
    keypoints0=keypoints1,
    keypoints1=keypoints3,
    matches=matches13,
    ax=ax[1, 0],
)
ax[1, 0].axis('off')
ax[1, 0].set_title(
    "Original Image vs. Transformed Image\n" "(all keypoints and matches)"
)
plot_matched_features(
    img1,
    img2,
    keypoints0=keypoints1,
    keypoints1=keypoints2,
    matches=matches12[::15],
    ax=ax[0, 1],
    only_matches=True,
)
ax[0, 1].axis('off')
ax[0, 1].set_title(
    "Original Image vs. Flipped Image\n" "(subset of matches for visibility)"
)
plot_matched_features(
    img1,
    img3,
    keypoints0=keypoints1,
    keypoints1=keypoints3,
    matches=matches13[::15],
    ax=ax[1, 1],
    only_matches=True,
)
ax[1, 1].axis('off')
ax[1, 1].set_title(
    "Original Image vs. Transformed Image\n" "(subset of matches for visibility)"
)
plt.tight_layout()
plt.show()
```

SURF

```matlab
imageFile = 'Tr-me_1286.jpg';
inputImage = imread(imageFile);
if size(inputImage, 3) == 3
    inputImageGray = rgb2gray(inputImage);
else
    inputImageGray = inputImage;
end
imshow(inputImageGray);
disp('Running SURF Detection in MATLAB...');
tic;
points = detectSURFFeatures(inputImageGray);
execTime = toc;
fprintf('Found %d SURF interest points in %f seconds.\n', points.Count, execTime);
hold on;
plot(points.selectStrongest(50));
hold off;
%https://www.mathworks.com/help/gpucoder/ug/feature-extraction-using-surf.html
```

ORB

```matlab

I = imread('Tr-me_1286.jpg');
if size(I, 3) == 3
    I = rgb2gray(I);
end
points = detectORBFeatures(I);
imshow(I);
hold on;
plot(points.selectStrongest(20));
%https://www.mathworks.com/help/vision/ref/orbpoints.html
```

References:

https://scikit-image.org/docs/0.25.x/auto_examples/features_detection/plot_sift.html

https://www.mathworks.com/help/gpucoder/ug/feature-extraction-using-surf.html

https://www.mathworks.com/help/vision/ref/orbpoints.html