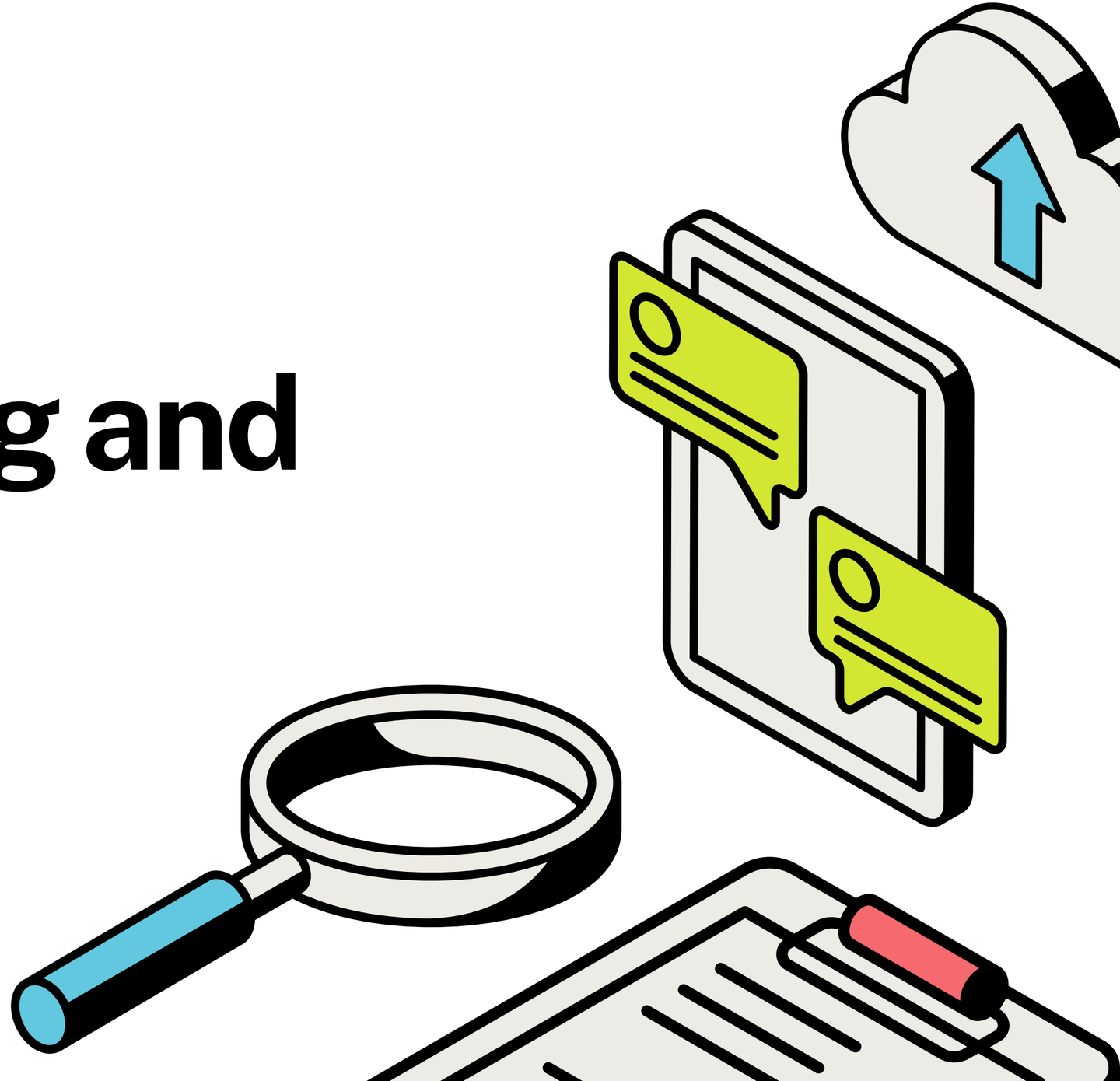
e-KTP Preprocessing and Recognition



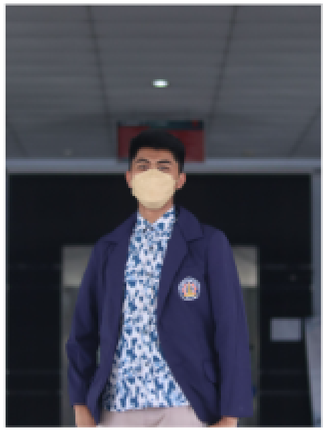
Date: November 30, 2023

Prepared by: Kelompok 4

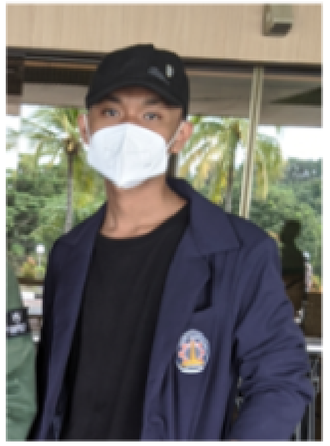
Who We Are

Muhammad Ali Zulfikar
Si Paling Hustler

Ulfi Mustatiq Abidatul Izza
Si Paling Hipster



Ilham Yudantyo
Si Paling Hekerr



M Izamul Fikri
Si Paling Hipster

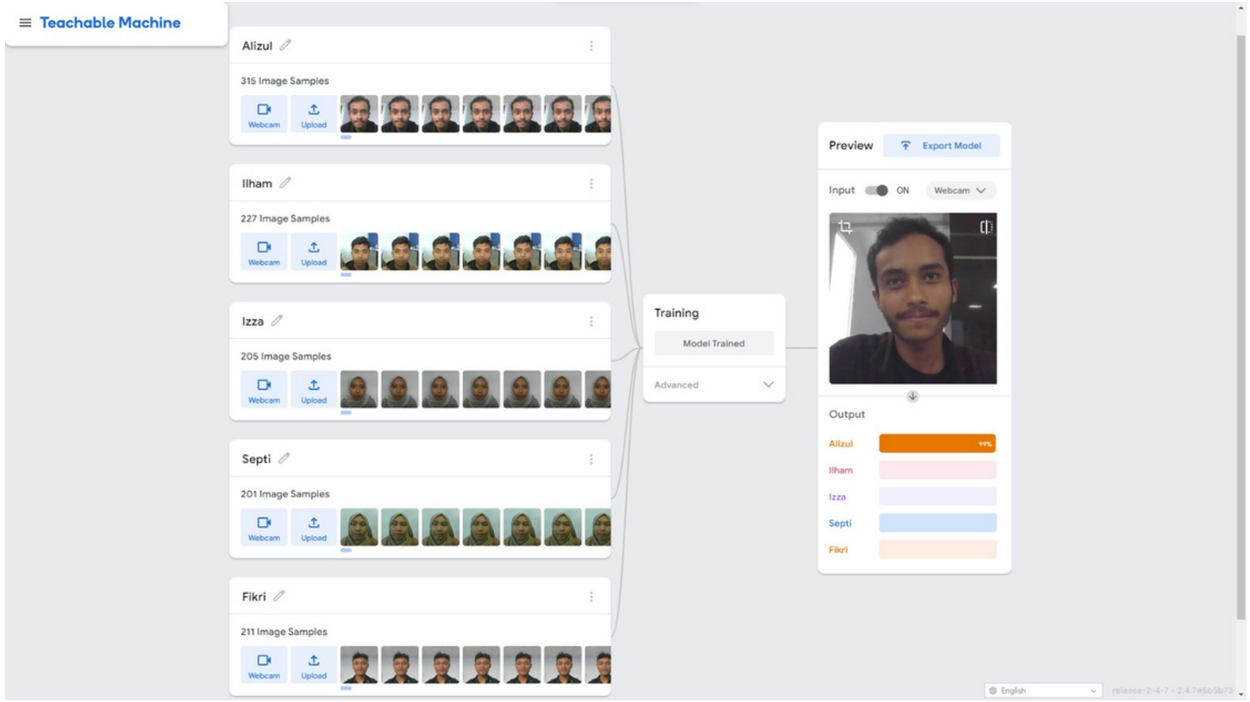
Septi Lutfiana
Si Paling Hipster

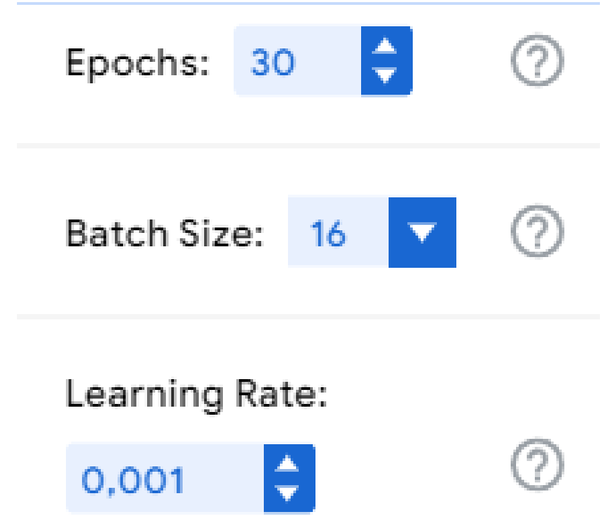
Section #1

Dataset Gathering

Kelompok 4
November 30, 2023

Dataset Gathering

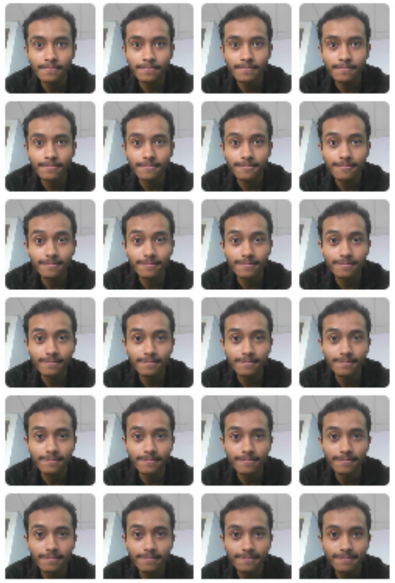




Using Teachable Machine, each dataset contains 200 images

Dataset Labeling

315 Image Samples

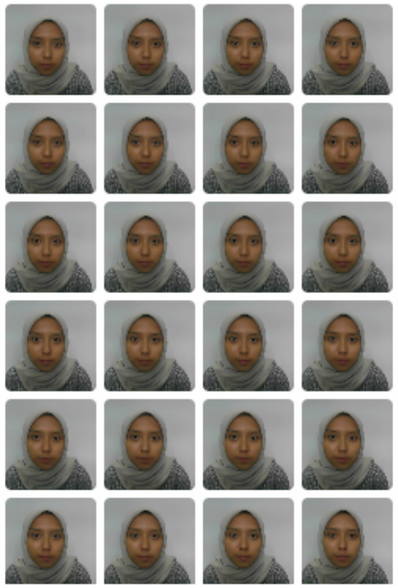


Alizul

227 Image Samples

Ilham

205 Image Samples

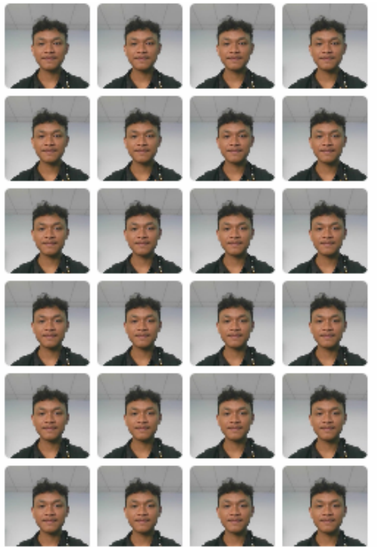


Izza

201 Image Samples

Septi

211 Image Samples



Fikri

Section #2

Library That We Used

Kelompok 4
November 30, 2023

Library

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import os
import numpy as np
from PIL import Image
from sklearn.pipeline import make_pipeline

from google.colab.patches import cv2_imshow
from google.colab import drive

from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np
```

Section #3

PreProcessing

Melakukan proses normalisasi dan langkah prapengolahan pada sampel gambar.

Kelompok 4
November 30, 2023

Pre Process Image

```python
def display_images(images, titles):
    fig, axs = plt.subplots(1, 5, figsize=(20, 20))
    for i in range(5):
        axs[i].imshow(images[i], cmap='gray')
        axs[i].set_title(titles[i])
        axs[i].axis('off')
    plt.show()
```

Menampilkan gambar dalam **satu baris dengan lima kolom**

```python
def preprocess_image(file_path):
    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
    image = Image.open(file_path).convert("RGB")
    size = (224, 224)
    image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)
    image_array = np.asarray(image)
    normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1
    data[0] = normalized_image_array
    return data, image_array
```

Mengonversi gambar yang sudah di-resample ke dalam bentuk **array numpy dan normalisasi array gambar ke dalam rentang -1 hingga 1**

Face Detection

```python
def face_detection(img, scaleFactor=1.1, minNeighbors=5, classifier='haarcascade_frontalface_alt.xml'):
    """
    Detect faces in an image using Haar cascade classifier.

    Parameters:
    - img: Input image
    - scaleFactor: Parameter specifying how much the image size is reduced at each image scale
    - minNeighbors: Parameter specifying how many neighbors each candidate rectangle should have to retain it
    - classifier: Path to the Haar cascade classifier XML file

    Returns:
    - img_rectangle: Image with rectangles drawn around detected faces
    """
    img_rectangle = img.copy()
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    haar_cascade = cv2.CascadeClassifier(classifier)

    if haar_cascade.empty():
        raise ValueError("Haar cascade classifier not loaded successfully.")

    rectangles = haar_cascade.detectMultiScale(img_gray, scaleFactor, minNeighbors)

    for (x, y, w, h) in rectangles:
        cv2.rectangle(img_rectangle, (x, y), (x+w, y+h), (0, 255, 0), 3)

    return img_rectangle
```

Crop Face Image

```python
def crop_face(img, scaleFactor=1.1, minNeighbors=5, classifier='haarcascade_frontalface_alt.xml', padding_y=80):
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    haar_cascade = cv2.CascadeClassifier(classifier)

    if haar_cascade.empty():
        raise ValueError("Haar cascade classifier not loaded successfully.")

    rectangles = haar_cascade.detectMultiScale(img_gray, scaleFactor, minNeighbors)

    if len(rectangles) == 0:
        raise ValueError("No face detected in the image.")

    (x, y, w, h) = rectangles[0]
    y = max(0, y - padding_y)
    h = min(img.shape[0] - 1, h + 2 * padding_y)
    cropped_face = img[y:y+h, x:x+w]

    return cropped_face
```

Localization and Segmentation

Input Data

5 Images

Face Recognition

Cropped Image

Section #3

Recognition

Mengidentifikasi dan pengenalan pada objek, pola, atau entitas pada sampel gambar dengan menerapakan machine learning.

Kelompok 4
November 30, 2023

Recognition Process

```python
def predict_image(model, image_data, class_names):
    prediction = model.predict(image_data)
    index = np.argmax(prediction)
    class_name = class_names[index]
    confidence_score = prediction[0][index]
    return class_name, confidence_score
```

Predict Image

*Melakukan **prediksi** terhadap gambar yang disediakan serta memberikan **confidence score***

Predict Image

Proses dalam prediksi dan lebeling sesuai dengan hasil prediksi model

```python
image_dir = './image/'

for filename in os.listdir(image_dir):
    if filename.endswith(".jpg"):
        file_path = os.path.join(image_dir, filename)
        image_data = preprocess_image(file_path)
        class_name, confidence_score = predict_image(model, image_data, class_names)

        print(f"File: {filename}")
        print(f"Predicted Class: {class_name}")
        print(f"Confidence Score: {confidence_score}")
        print("\n")
```

Recognition Process

```
1/1 [==============================] - 2s 2s/step
File: 2.jpg
Predicted Class: 1 Ilham

Confidence Score: 0.6260475516319275


1/1 [==============================] - 0s 46ms/step
File: 3.jpg
Predicted Class: 3 Septi

Confidence Score: 0.9004600048065186


1/1 [==============================] - 0s 40ms/step
File: 0.jpg
Predicted Class: 0 Alizul

Confidence Score: 0.9405743479728699


1/1 [==============================] - 0s 45ms/step
File: 1.jpg
Predicted Class: 0 Alizul

Confidence Score: 0.9402249455451965


1/1 [==============================] - 0s 48ms/step
File: 4.jpg
Predicted Class: 2 Izza

Confidence Score: 0.9948586225509644
```

Hasil Prediksi

Predicted Class: 0 Alizul

Confidence Score: 94.06%

Predicted Class: 0 Alizul

Confidence Score: 94.02%

Predicted Class: 1 Ilham

Confidence Score: 62.60%

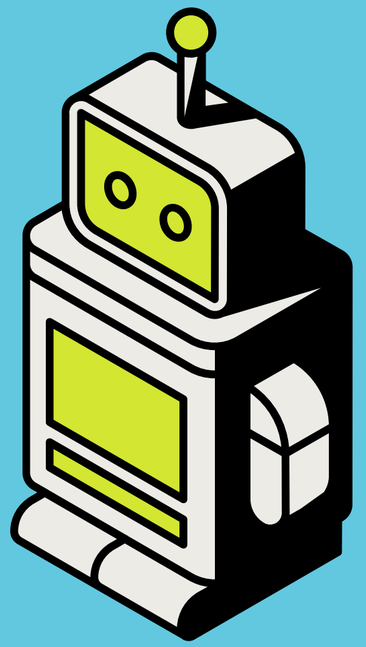
Predicted Class: 3 Septi

Confidence Score: 90.05%

Predicted Class: 2 Izza

Confidence Score: 99.49%

Thank you.



Kelompok 4
November 30, 2023