Milestone Report # 1

Face Recognition Project

Group H

- Saurabh Laltaprasad Gangwar
- Brayan Leonardo Gil Guevara
- Rohit Kumar
- Marzieh Mohammadi Kokaneh
- Eduardo Roberto Williams Cascante

Contents

Introduction 3
Progress Report 4
Data collection 4
Data source 4
Data preparation 4
Data annotation 4
Labelme 4
Data labeling 5
Modified Timeline 9
Next Steps 10
Challenges faced 11
Lessons Learned 12
Conclusion 13
References 14

Introduction

The face recognition industry has been crucial for security purposes over the years, however, once COVID hit in late 2019, more requirements for this technology have been raised.

This document explores the use of deep learning for facial recognition. The Idea comes from creating a local dataset having different photos of people on a team and training the model to detect the identity of each person. We will discuss how to collect and prepare data, as well as develop models for this combined task. The target for this project is security companies for letting people to entrance the office or not!

Progress Report

Data collection

Data source

Since the goal of our project is to build a model which tries to recognize our faces, we took one hundred pictures (each group's member) from our cellphone's personal gallery. This set of pictures was made up as follows: ninety pictures with our faces in first plane and ten pictures without our faces. Those ten pictures were intended not to be labeled.

Data preparation

For the data preparation process, we performed two main steps: cropping and resizing.

The purpose of cropping was to set pictures in an exactly square shape, it means, with the same height and width number of pixels. This was necessary for the resizing step. In addition, we first cropped them to make faces take up a third of the photo at least, in other words, we cropped elements in the pictures other than our faces.

So as to reduce pictures weight and not to collapse the following processes (annotation, augmentation, etc.) we had to resize all the pictures to 480 x 480 pixels. Additionally, augmentation process needs all photos with the same shape because it performs a little random cropping and also receive the normalized labels coordinates as input. These coordinates are strictly related to pictures width and height.

Both cropping and resizing steps were made manually by using Paint and Adobe Photoshop.

Data annotation

Labelme

We performed data annotations by using a tool called Labelme. We set up and ran Labelme from a python notebook, with the following commands:

- **%pip install labelme**
- **!labelme**

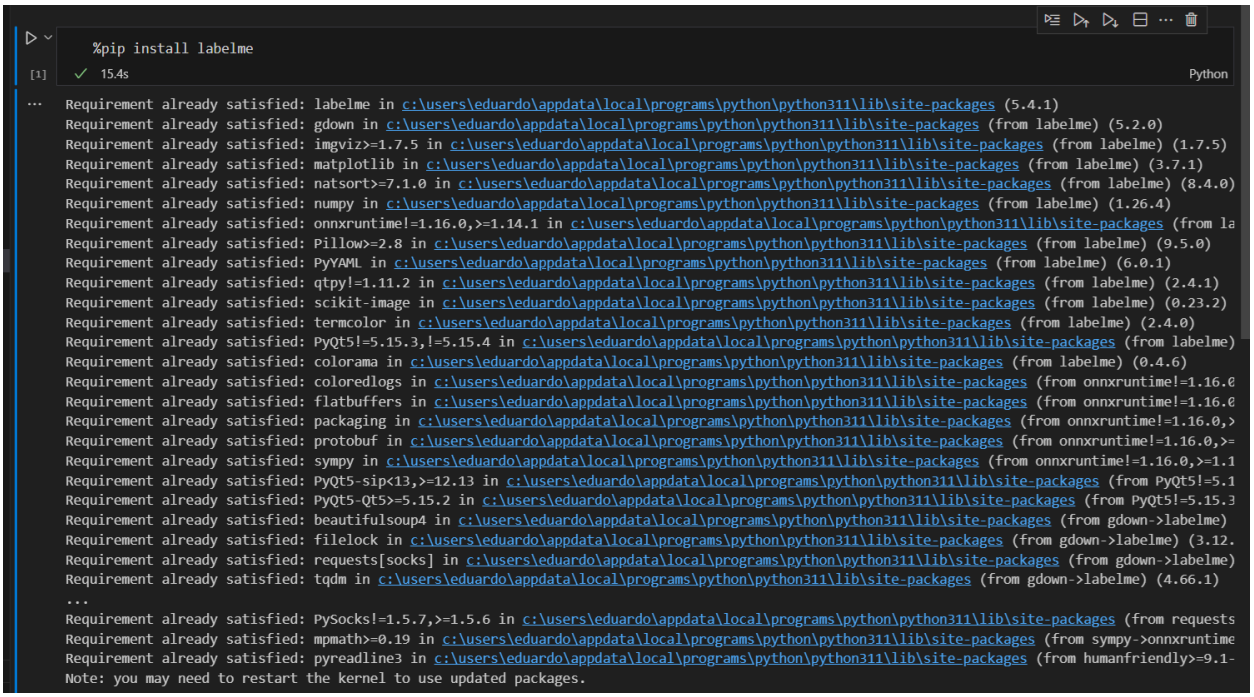

Figure 1 Labelme installation

Data labeling

In order to label correctly each face in pictures we defined some settings in Labelme:

- **Open Dir:** It refers to the folder which contains all cropped and resized photos.

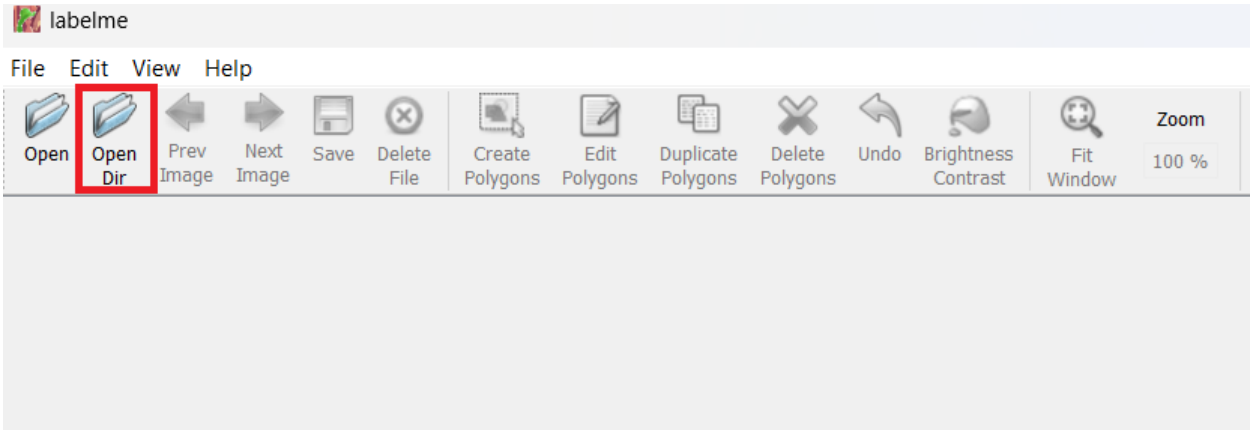

Figure 2 Labelme – Open Dir

- **Output Dir:** It refers to the folder where Labelme will save labeling information files. They are files in json format which contain information related to the coordinates of labeling as well as classes names, etc.

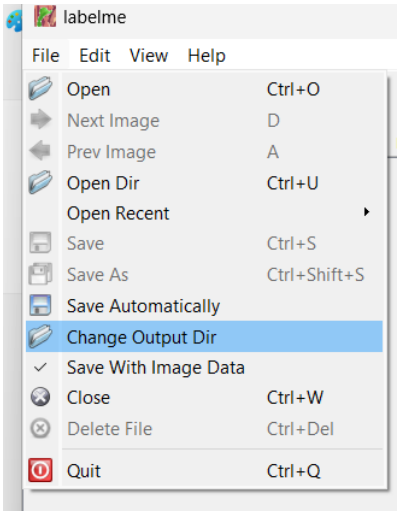

Figure 3 Labelme – Open Dir

- **Saving mode:** Additionally, we setup Labelme for it to automatically save each label we did, instead of pressing Save button for each file.

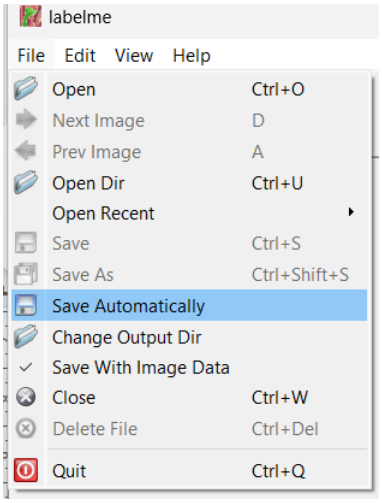

Figure 4 Labelme – Saving mode

- **Shape:** Finally, we selected the shape (rectangle) we wanted to draw for labeling.

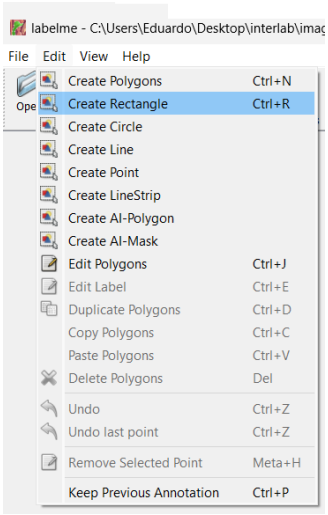

Figure 5 Labelme – Shape

Once Labelme was set up, the next step was to label faces on each photo, as shown in the next figure:

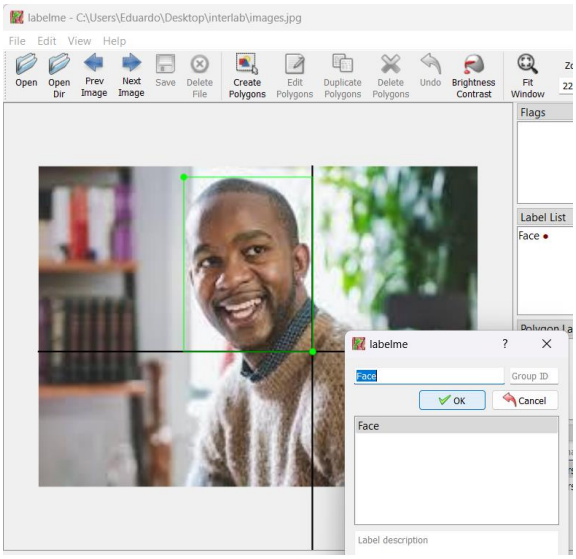

Figure 6 Labelme – Labeling

For each labeling we did Labelme saved the related data in a json file with the same name as the picture file. This file contains some information such as version, picture filename, image weight, image height, however the most important information are the label and the points, this latter contains the coordinates of labeling.

```
{
  "version": "5.4.1",
  "flags": {},
  "shapes": [
    {
      "label": "Face",
      "points": [
        [
          86.95851528384279,
          6.4803493449781655
        ],
        [
          164.25109170305674,
          111.06550218340612
        ]
      ],
      "group_id": null,
      "description": "",
      "shape_type": "rectangle",
      "flags": {},
      "mask": null
    }
  ],
  "imagePath": "images.jpg",
  "imageData": "/9j/4AAQSkZJRgABAQAAAQABAAD
  "imageHeight": 480,
  "imageWidth": 480
}
```


Figure 7 Labelme – json file

Modified Timeline

The timeline does not present any update after being submitted for project proposal. The deadlines have been completed as promised:

- Project Proposal (May 21st) - **Delivered**

  Request proposal Project

- Milestone 1 (June 4th) - **Delivered**

  Data collection technique
  Data annotation process

- Milestone 2 (June 18th)

  Data splitting process
  Data augmentation process

- Milestone 3 (July 2nd)

  Neural net training process

- Milestone 4 (July 16th)

  Neural Net testing process

- Milestone 5 (July 30th)
  Deployment documentation

- Final Report (Aug 13th)

Next Steps

The upcoming stages of the project rely on the below steps:

Neural Net model building

Selecting of the model for the Neural Network Architecture suitable for face recognition. There are some popular options like CNNs (Convolutional Neural Networks) like VGGFace or ResNet. Also a part of this step is the hyper parameter tuning, defining and optimizing the number of layers, number of filters and learning rate.

Classification / Regression

This will be clearly a classification task, but including the challenge of making a differentiation of every individual.

Convolutional Neural Net

The model is being implemented with some of the most familiar libraries for deep learning like TensorFlow, PyTorch or Keras.

Loss function

Selecting the most appropriate loss function to evaluate the performance of the model during training is a key step. For classification tasks some of the most common options is cross-entropy loss.

Optimizer function

An optimizer function will be considered to modify the model's weights based on the loss from the previous stage. Some options to consider here are SGD with variants like Adam or RMSprop

GitHub repository update

This is also a very important step as focusing on the code versioning and code uploads will help establish a version control in GitHub

Project board update

As decided to work with ClickUp, the task breakdown and the progress tracking will help ensure that all deadlines are monitored as well as every member has tasks assigned.

Challenges faced

The main challenge faced so far has been related to the dataset option chosen. For this project, the dataset set has been built by the members itself, as every individual has shared initially 100 pictures that will be part of the training and test stages of the project.

In detail, the challenge is related with choosing the right picture for creating the dataset, as per example a selfie where only 60% of the human face can be seen it is a condition that cannot be given to the model.

The second challenge came after choosing the photos, since now every picture needed to be processed in the same way (reduced to 480x480 in resolution) and cropping was individually done, since the aim was to put the focus the face to be recognized.

More challenges arise after completing the photos, since the members were not sure about the optimal size for the dataset or the number of photos. The workaround here was then try and error

The final challenge for this milestone came when the board realized that local compute power was not enough since most of the member are lacking dedicated GPUs with enough CUDA Cores. The alternative chosen here was to go with Google Colab, it made a significant difference but still the runtime will randomly disconnect. This latter challenge was solved when upgrading to a Colab Pro Subscription.

Lessons Learned

As many challenges has been faced for this particular milestone, this also means a significant number of lessons learnt, this will be all again review during the final report but it is important to talk about the dataset diversity, since relying on self-built datasets with limited variations can hinder the model performance.

Something else worth mentioning is the inconsistency in cropping methods that lead to unnecessary complexity. Moving forward a clear guideline for images preprocessing and a detailed standard for cropping and resizing approach for all the images in the dataset.

Also the trial-and-error approach for measuring the optimal dataset size was inefficient. To avoid this a preliminary experiment should be performed to meet the desired level of accuracy

Having a workstation or just access to one with appropriate GPUs might be necessary for additional compute power and can be an alternative to paying a Google Cloud Subscription, however, the price vs benefit of the current solution cannot be beat so far.

Conclusion

The first milestone of the face recognition project has been successfully completed, but not without valuable learning experiences. Successfully built a self-sourced dataset and addressed challenges related to image selection, processing, and computational limitations.

Here are the key takeaways from this milestone:

Dataset Considerations: While a self-built dataset provided a starting point, its limitations became evident. In the future, it is necessary to incorporate publicly available datasets to enhance diversity and model performance.

Data Preprocessing Standardization: Standardized data preprocessing procedures, including cropping and resizing, will be crucial for future projects to ensure consistency and streamline workflows.

Scalability Planning: Adopting a more structured approach to determining optimal dataset size based on research or preliminary experiments.

Collaborative Cloud Resources: For computationally intensive tasks, exploring cloud-based solutions with guaranteed uptime or investing in member workstations with appropriate GPUs might be necessary.

These lessons learned will guide us as we move forward to the next exciting phase of the project: building and training the neural network model for face recognition. The board will delve into selecting an appropriate architecture, setting hyperparameters, and implementing the model using deep learning libraries. Additionally, a robust version control system with Git and GitHub will be established, and utilize project management tools to track progress and ensure efficient collaboration.

The board remains confident that the knowledge gained from this first milestone will pave the way for the development of a robust and effective face recognition system.

References

CS231n convolutional neural networks for visual recognition. (n.d.). Retrieved June 11, 2024, from https://cs231n.github.io/

Géron, A. (2019). *Hands-On machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Ng, A. (n.d.). *Deep learning*. Coursera. Retrieved June 11, 2024, from https://www.coursera.org/specializations/deep-learning

Scikit-Learn (2024). *Official Documentation.* Retrieved June 11, 2024 from https://scikit-learn.org/stable/

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.