DEEP FACE RECOGNITION

BDM 3035 - BIG DATA CAPSTONE PROJECT

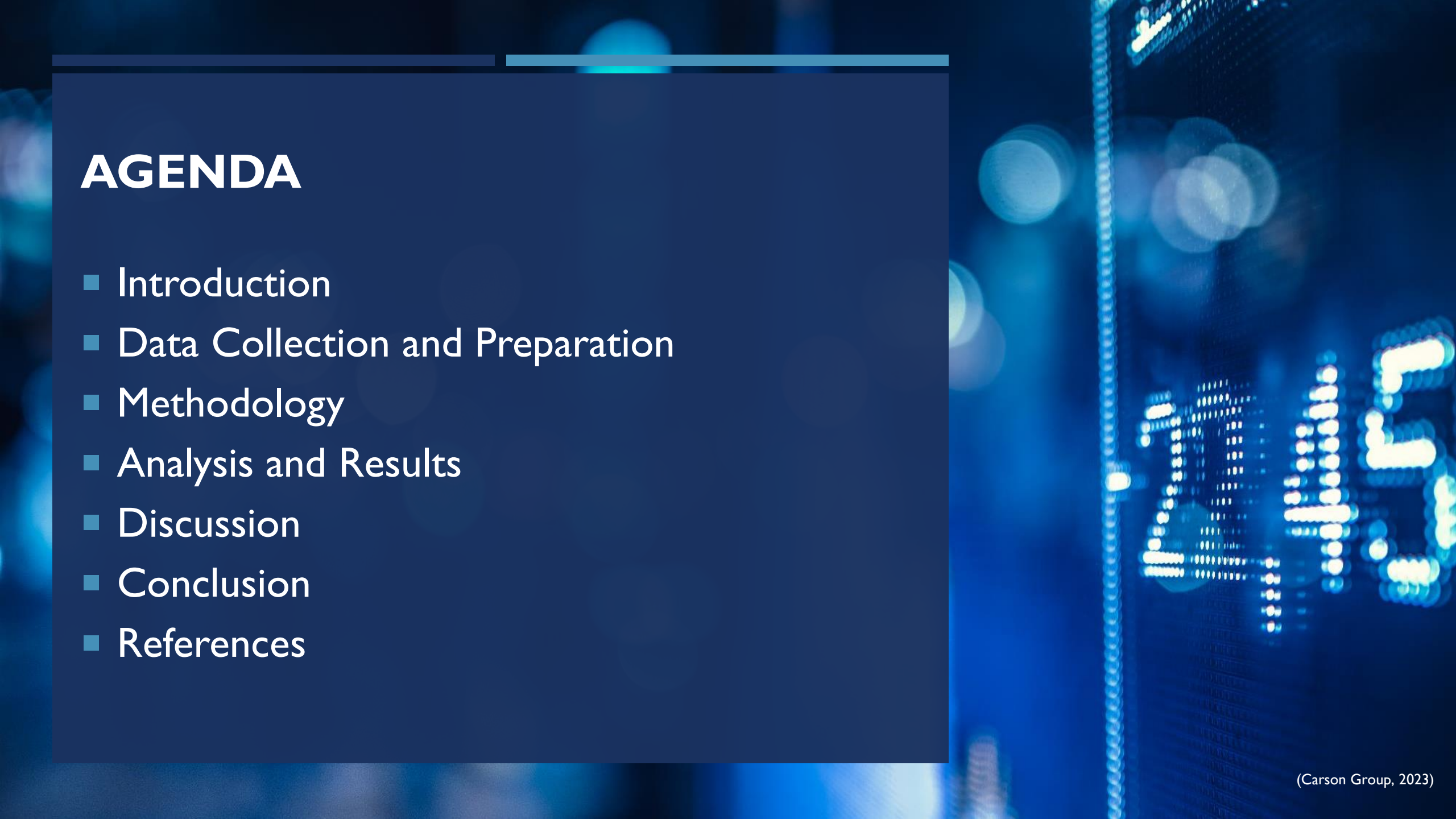
GROUP H

(VentureBeat, 2024)

- Marzieh Mohammadi Kokaneh (C089839)
- Saurabh Laltaprasad Gangwar (C0894380)
- Brayan Leonardo Gil Guevara (C0902422)
- Rohit Kumar (C0895100)
- Eduardo Williams Cascante (C0896405)

AGENDA

- Introduction
- Data Collection and Preparation
- Methodology
- Analysis and Results
- Discussion
- Conclusion
- References



(Carson Group, 2023)

INTRODUCTION

- This project explores the use of deep learning techniques for facial recognition, useful for understanding how Convolutional Neural Networks works
- This project can be applied in several fields such as:
  - Biometric control and access
  - Video surveillance solutions
  - Lost people searching, and so on
- The original idea comes from creating a local dataset having different photos of our team's members and training the model to detect the identity of each person. However, due some resources limitations (time, processing capacity, etc.) we reduced the scope to just two group's members.

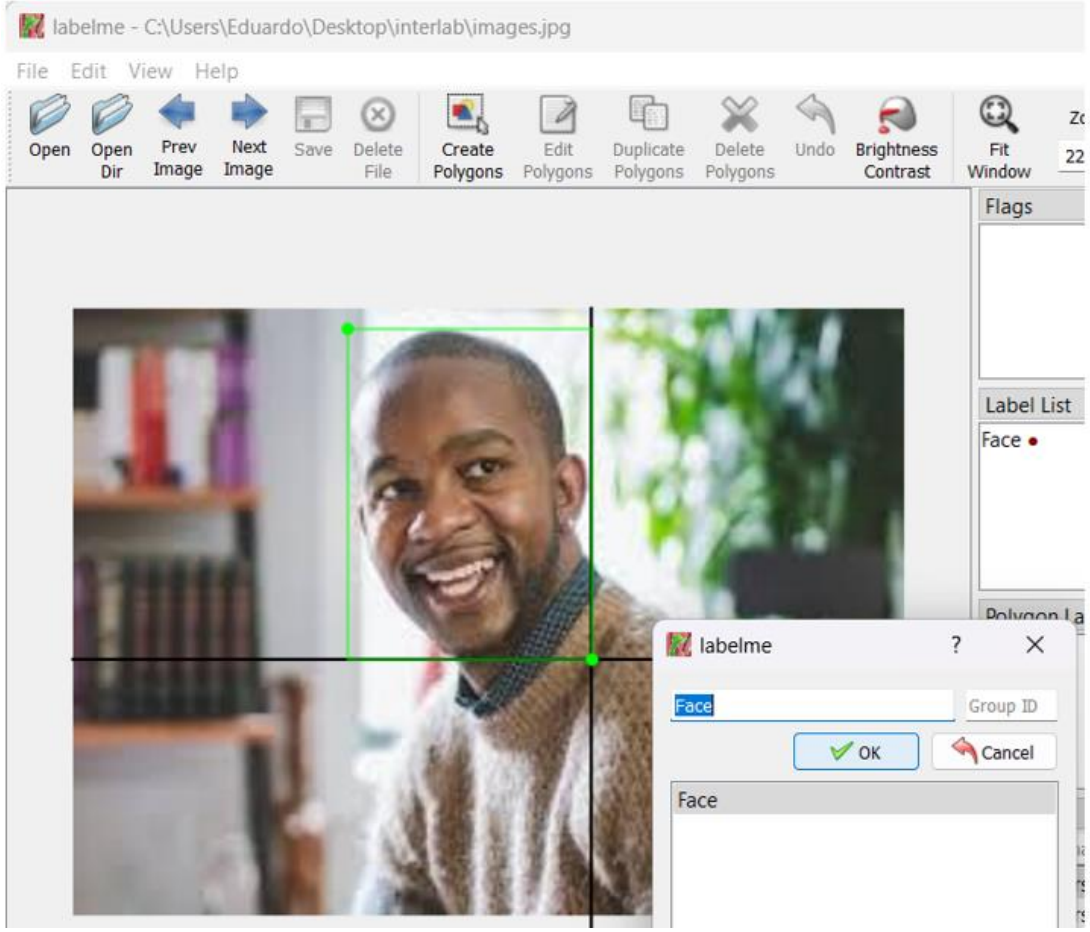
(Cline, 2024)

DATA COLLECTION AND PREPARATION

- Data collection
  - Each person took 150 pictures from its smartphone and fixed them to 480 x 480 pixels

- Data splitting
  - Training dataset → 100 pictures
  - Validation dataset → 20 pictures
  - Test dataset → 30 pictures

DATA COLLECTION AND PREPARATION

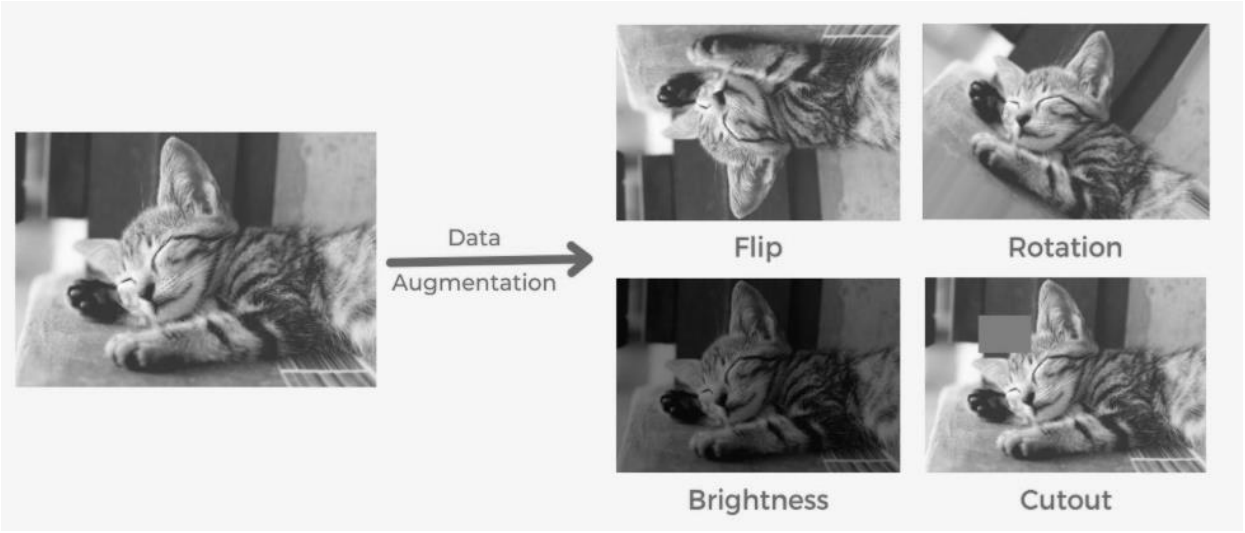


- Data annotation

  - LabelMe is a free graphical annotation tool for image and video data
  - LabelMe allows to label different objects over the same picture
  - LabelMe saves labels and coordinates in a Json file (for each picture)

DATA COLLECTION AND PREPARATION

- Data augmentation
  - Since 150 pictures are not enough for this process, we had to multiply them
  - Albumentations is a fast and flexible image augmentation library
  - Albumentations not only multiplied pictures but also applied some transformations on their position, heights, widths, and number of color channels, in order for training process to learn about different faces conditions



(Panchuang, 2021)

METHODOLOGY

- We created a CNN (Convolutional Neural Network), by using Tensorflow, Keras, GlobalMaxPooling2D, VGG16, among others

  - Input layer shape (120,120,3) → (height, width, RGB)
  - VGG16 layer
  - GlobalMaxPooling2D layer for Classification → 2048 units
  - GlobalMaxPooling2D layer for Regression → 2048 units
  - Output layer for Classification → **Classes**
  - Output layer for Regression → **Coordinates**

METHODOLOGY

- TensorFlow
  - It is an open-source machine learning framework developed by Google.
  - It's widely used for building and deploying machine learning models, particularly deep learning models.

- Keras
  - It is a high-level neural networks API, written in Python and capable of running on top of TensorFlow.

METHODOLOGY

- VGG16
  - It is a convolutional neural network (CNN) architecture that was proposed by the Visual Geometry Group (VGG) at the University of Oxford.
  - It is one of the most well-known deep learning models for image classification.

- GlobalMaxPooling2D
  - It is a pooling operation commonly used in convolutional neural networks.
  - It reduces the spatial dimensions (height and width) of the input while retaining the most important features.

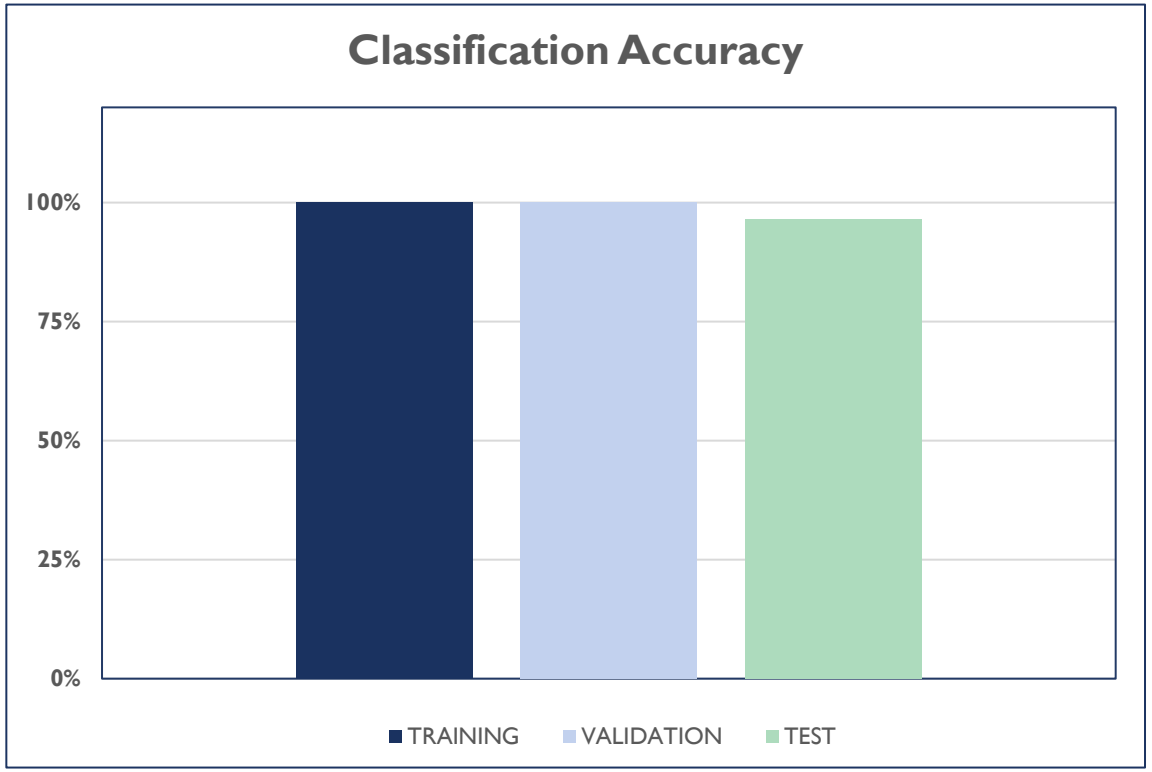
METHODOLOGY

- For compiling the model, we used the following settings:
  - Optimizer → Adam
  - Classification loss → SparseCategoricalCrossentropy

- For training process, we used the following callbacks:
  - TensorBoard → Just for logging (not for visualizations)
  - EarlyStopping → It stops epochs if metrics no longer get better
  - ModelCheckpoint → For tracking and saving the best weights and metrics

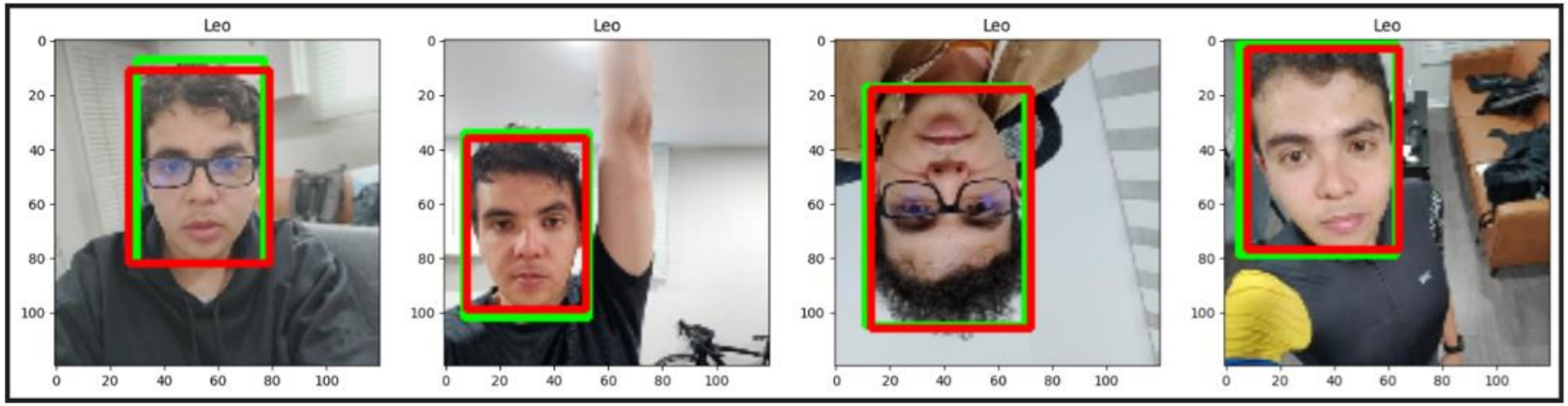
ANALYSIS AND RESULTS

(VentureBeat. 2024)



ANALYSIS AND RESULTS

- To evaluate regression predictions (coordinates) we used Intersection Over Union metric
- Intersection over Union (IoU) is used to measure the accuracy of object detection by comparing the ground-truth bounding box (green) to the predicted bounding box (red)
- We got 89.94% of precision for this metric



DISCUSSION

- Regarding to classification predictions, we got 100% of accuracy in train and validation datasets, however we also got 96.53% in test dataset which is good and demonstrate that our model is not overfitted.

- As for regression predictions, we used Intersection Over Union metric in test dataset, and we got 89.94%. Let's consider that any value greater than 0.5 on IoU is typically a good score.

DISCUSSION

- We got 78% of classification accuracy on an unseen dataset, it means the accuracy decreased.

- However, we understand this could be due to factors such as: unseen pictures was taken with other phone model, some faces were not exactly in front of the camera, people appeared pulling faces in some pictures or faces covered just a little percentage of the picture, unlike training dataset where faces covered at least 30% of pictures.

- As a learning, we must select pictures from different sources and different conditions, such as: positions, light exposition, etc., to improve the robustness of the model.

CONCLUSION

- Our project aimed to develop a robust face recognition model using deep learning techniques, specifically Convolutional Neural Networks (CNNs).
- Through a series of meticulously planned steps, from data collection and preprocessing to model training and evaluation, we achieved significant milestones.
- Comprehensive data preprocessing and augmentation, which enhanced the model's ability to generalize.
- Our project successfully developed a deep learning-based face recognition model with high accuracy. The use of CNNs and transfer learning significantly contributed to the model's performance by reduced training time and improved performance.

FINAL STEPS

- DEMO
- QUESTIONS

REFERENCES

1. Carson Group. (2023, October 3). *Four reasons we believe stocks won't crash in October*. https://www.carsongroup.com/insights/blog/four-reasons-we-believe-stocks-wont-crash-in-october/
2. Cline, R. (2024, March 18). *Biometric access control systems: Your complete guide*. ButterflyMX® - Video Intercoms & Access Control Systems. https://butterflymx.com/blog/biometric-access-control/
3. CS231n: *Convolutional neural networks for visual recognition*. (n.d.). Retrieved June 11, 2024, from https://cs231n.github.io/
4. Géron, A. (2019). *Hands-on machine learning with scikit-learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
5. Ng, A. (n.d.). *Deep learning*. Coursera. Retrieved June 11, 2024, from https://www.coursera.org/specializations/deep-learning
6. Panchuang. (2021, July 12). *Image augmentation in deep learning*. Panchuang Blog. https://panchuang.net/2021/07/12/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E4%B8%AD%E7%9A%84%E5%9B%BE%E5%83%8F%E5%A2%9E%E5%BC%BA%E7%AE%80%E4%BB%8B/
7. Scikit-Learn. (2024). *Official Documentation*. Retrieved June 11, 2024, from https://scikitlearn.org/stable/
8. tensorflow.org. (n.d.). *Get started with TensorBoard*. TensorFlow. Retrieved from https://www.tensorflow.org/tensorboard/get_started
9. VentureBeat. (2024, August 9). *The art and science of SaaS pricing: True usage-based pricing*. https://venturebeat.com/business/the-art-and-science-of-saas-pricing-true-usage-based-pricing/

THANKS!