



Misk مسک
مؤسسة محمد بن سلمان
Mohammed Bin Salman
Foundation

Data Science Immersive 2022-01
Capstone Project

August 25th, 2022

REAL-TIME ARABIC SIGN LANGUAGE INTERPRETER

Prepared by:
Sara Al-Qahtani

Instructors:
Dr. Rick Scavetta
Lujain Felemban

AGENDA

Introduction & Problem Statement

Dataset Description

Suggested Solution

Challenges + Troubleshooting

Future Work

Conclusion

INTRODUCTION

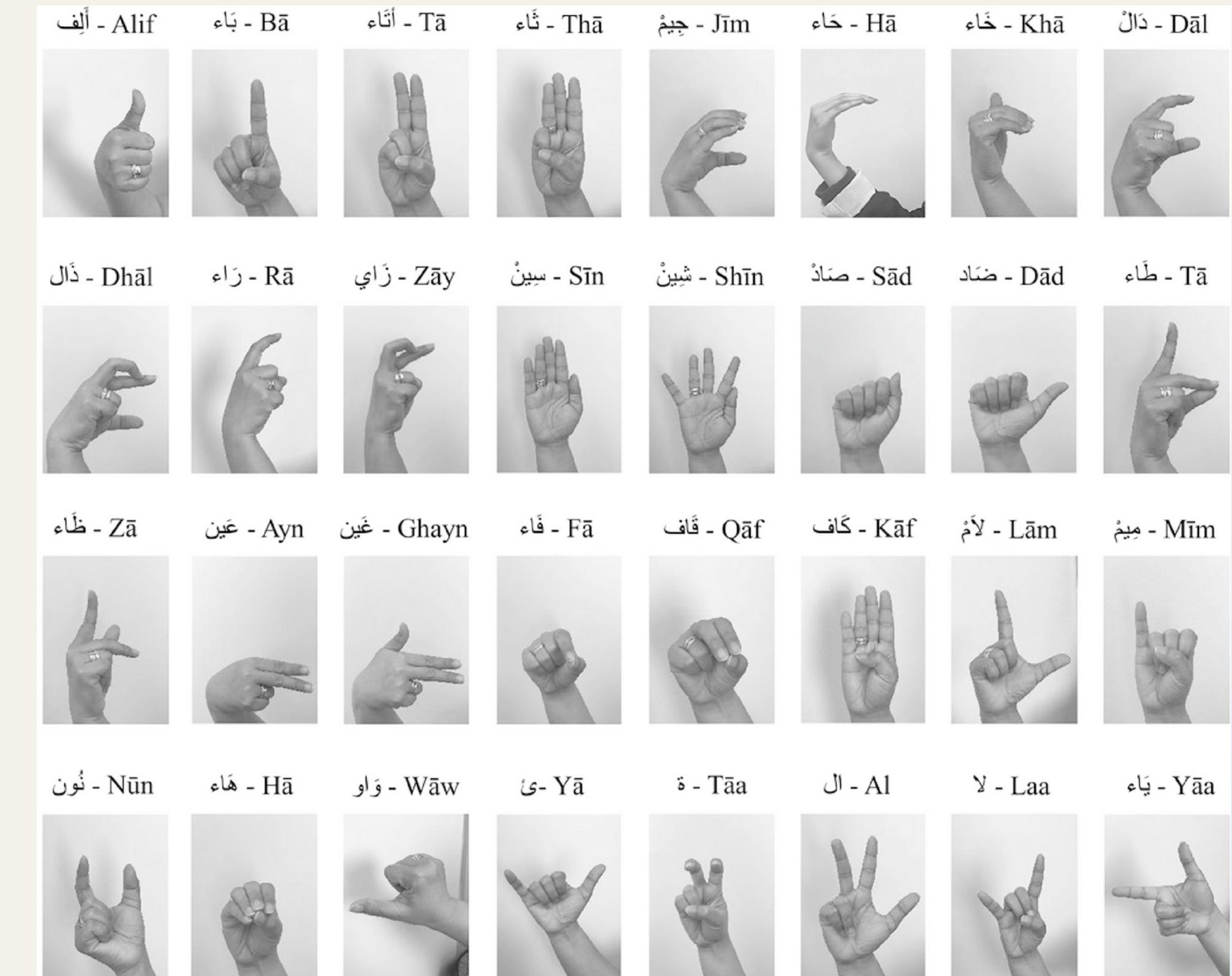
Problem statement

- Effective communication for deaf and mute people is very hard when they meet people who don't know or speak sign language. Pattern recognition and computer vision can be used to mitigate that issue and make communication easier for those minorities.
- Deep learning, specifically *transfer learning*, is used to recognize hand gestures and interpret their meaning and which letter they refer to in order to help deaf and mute people communicate with us and vice versa.
- **Transfer learning** consists of taking features learned on one problem, and leveraging them on a new, similar problem.

DATASET DESCRIPTION

The ArSL2018 is a new comprehensive fully labelled dataset of Arabic Sign Language images launched in Prince Mohammad Bin Fahd University, Al Khobar, Saudi Arabia to be made available for researchers in the field of Machine Learning and Deep Learning.

- The dataset contains **54,049** images and **32** letters/classes.



SUGGESTED SOLUTION

Steps:

- 1 - **Creating a label map.**
- 2 - Labeling images using LabelImg.
- 3 - Creating TensorFlow Records (tfrecords).
- 4 - Training the transfer learning model (MobileNet SSD V2)
- 5 - Fine-tuning and configuring of transfer model.

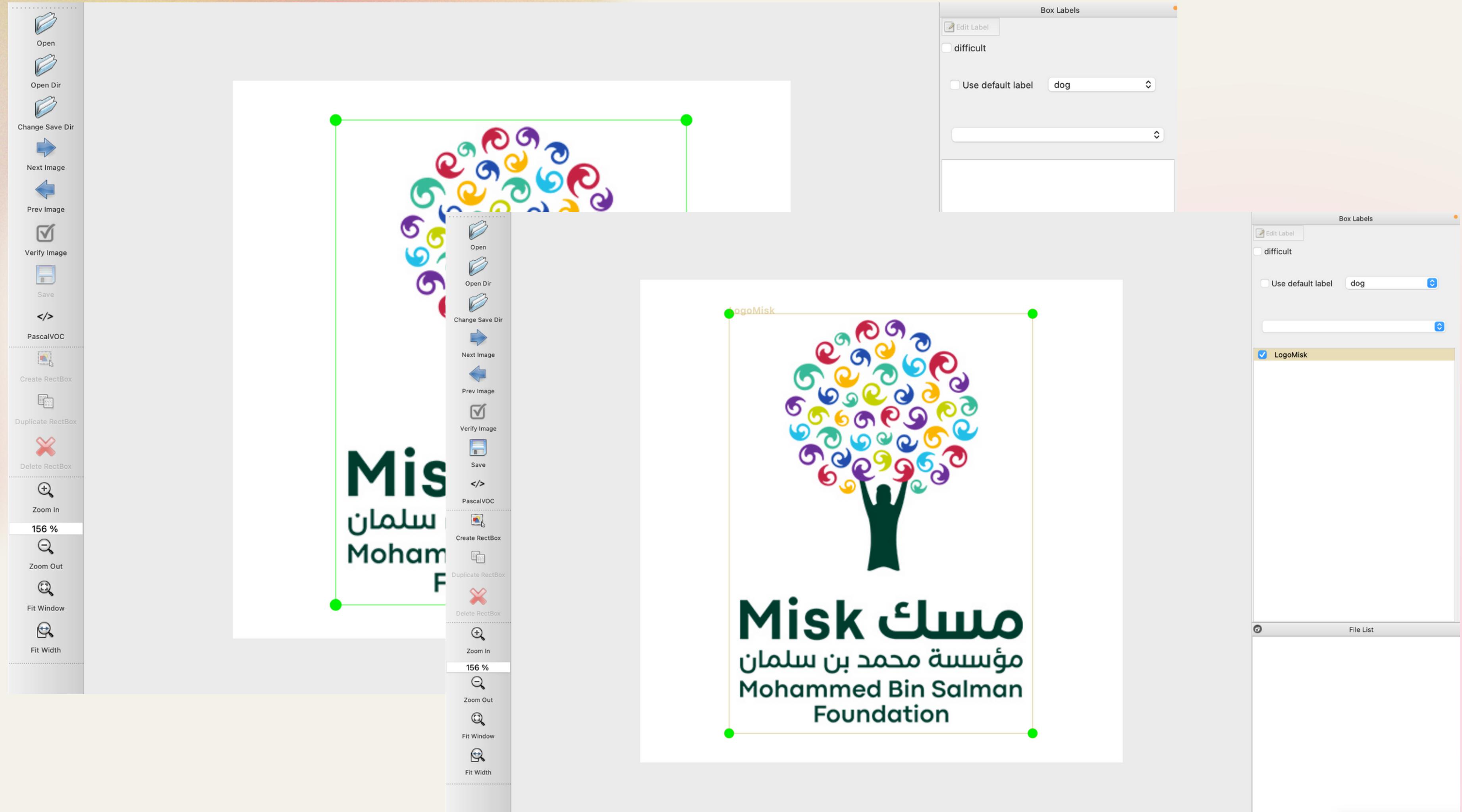
```
label_map.pbtxt •  
Tensorflow > workspace > annotations > label_map.pbtxt  
1 item {  
2   name: 'ا'  
3   id: 1  
4 }  
5 item {  
6   name: 'ب'  
7   id: 2  
8 }  
9 item {  
10  name: 'ج'  
11  id: 3  
12 }  
13 .  
14 .  
15 .  
16 .  
17 .  
18 .  
19 item {  
20   name: 'د'  
21   id: 30  
22 }  
23 item {  
24   name: 'ه'  
25   id: 31  
26 }  
27 item {  
28   name: 'ز'  
29   id: 32  
30 }  
31
```

SUGGESTED SOLUTION

Steps:

- 1 - Creating a label map.
- 2 - Labeling images using LabelImg.**
- 3 - Creating TensorFlow Records (tfrecords).
- 4 - Training the transfer learning model (MobileNet SSD V2)
- 5 - Fine-tuning and configuring of transfer model.





```
<annotation>
  <folder>Desktop</folder>
  <filename>feature-misk.jpg</filename>
  <path>/Users/saraali/Desktop/Screen Shot 2022-08-25 at 6.43.34 AM.png</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1970</width>
    <height>1376</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>LogoMisk</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>160</xmin>
      <ymin>349</ymin>
      <xmax>1765</xmax>
      <ymax>1105</ymax>
    </bndbox>
  </object>
</annotation>
```

SUGGESTED SOLUTION

Steps:

- 1 - Creating a label map.
- 2 - Labeling images using LabelImg.
- 3 - Creating TensorFlow Records (tfrecords).**
- 4 - Training the transfer learning model (MobileNet SSD V2)
- 5 - Fine-tuning and configuring of transfer model.

Only worked with Docker container running Python 3.6 and Tensorflow == 1.13.1

SUGGESTED SOLUTION

Steps:

- 1 - Creating a label map.
 - 2 - Labeling images using LabelImg.
 - 3 - Creating TensorFlow Records (tfrecords).
- 4 - Training the transfer learning model
(MobileNet SSD V2)**
- 5 - Fine-tuning and configuring of transfer model.

SUGGESTED SOLUTION

Steps:

- 1 - Creating a label map.
- 2 - Labeling images using LabelImg.
- 3 - Creating TensorFlow Records (tfrecords).
- 4 - Training the transfer learning model (MobileNet SSD V2)
- 5 - Fine-tuning and configuring the transfer model.**

Fine-tuned Model	
batch_size	32
total_steps	20k
learning_rate_base	↑ 0.01
warmup_learning_rate	↑ 0.056666

Loss

Loss/classification_loss
tag: Loss/classification_loss

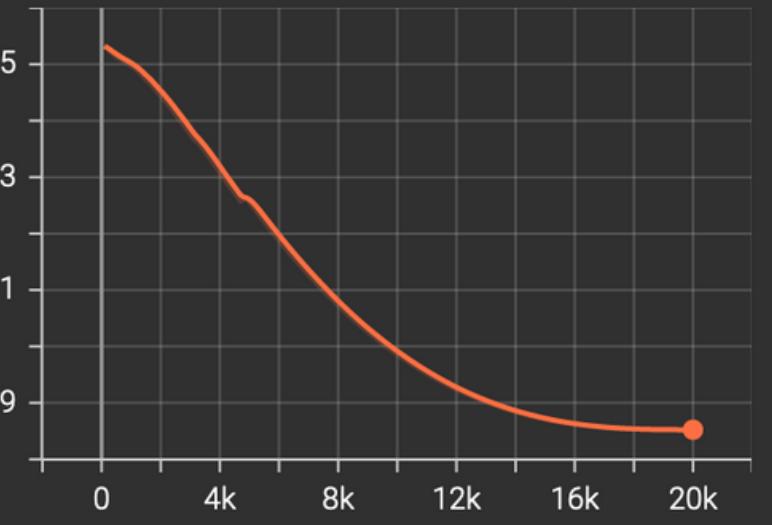


Loss/localization_loss
tag: Loss/localization_loss

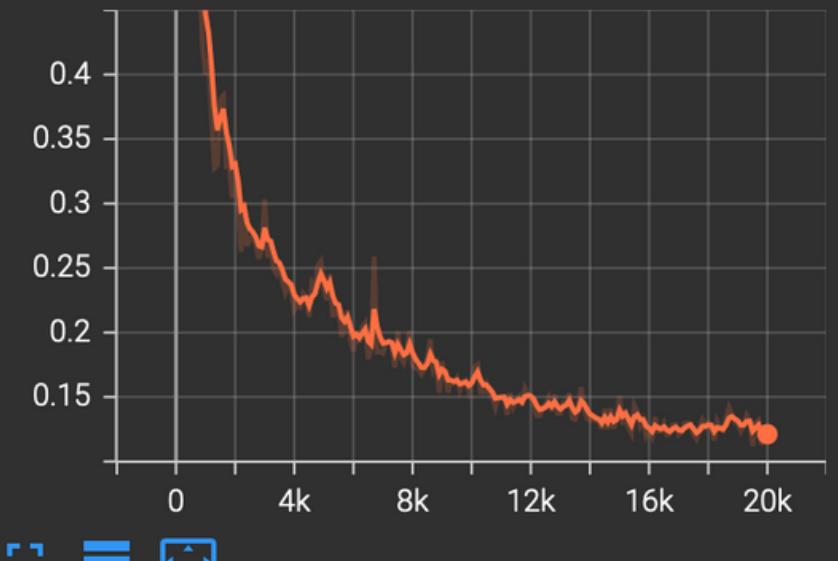
Loss/localization_loss
tag: Loss/localization_loss



Loss/regularization_loss
tag: Loss/regularization_loss



Loss/total_loss
tag: Loss/total_loss



Fine-tuned Model

32

20k

↑ 0.01

↑ 0.056666

Loss

Loss/classification_loss
tag: Loss/classification_loss



[] [] []

Loss/localization_loss
tag: Loss/localization_loss

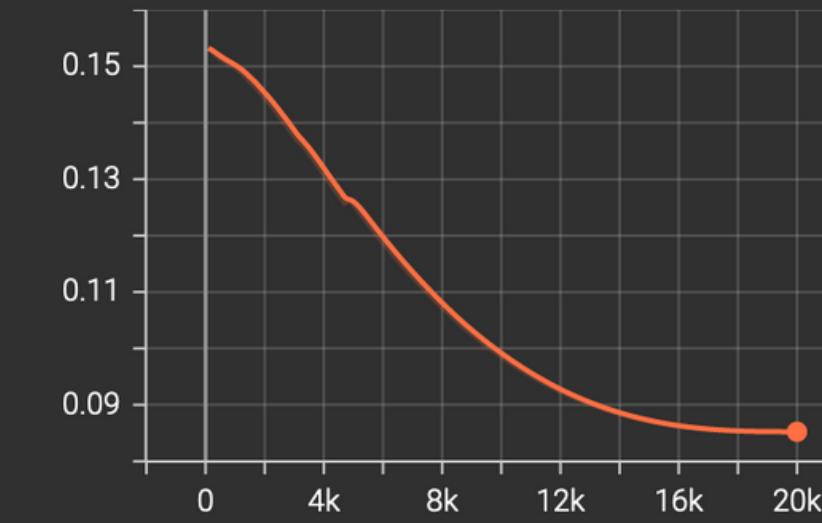
Loss/localization_loss
tag: Loss/localization_loss



[] [] []

Loss/regularization_loss
tag: Loss/regularization_loss

Loss/regularization_loss
tag: Loss/regularization_loss



[] [] []

Loss/total_loss
tag: Loss/total_loss

[] [] []



[] [] []

Fine-tuned Model

32

20k

0.01

0.056666



Loss

Loss/classification_loss
tag: Loss/classification_loss



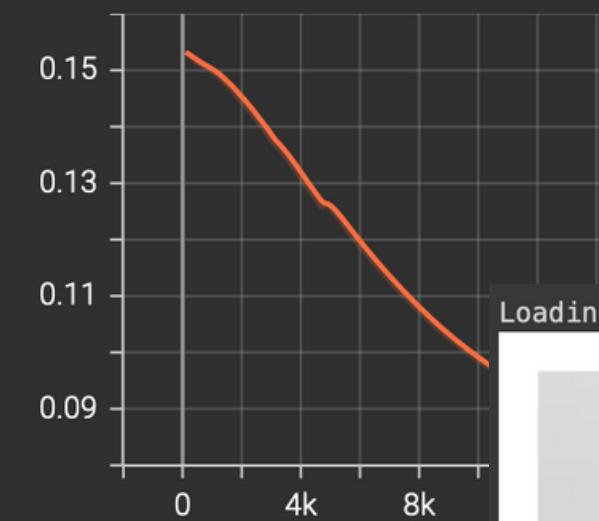
Loss/localization_loss
tag: Loss/localization_loss

Loss/localization_loss
tag: Loss/localization_loss



Loss/regularization_loss
tag: Loss/regularization_loss

Loss/regularization_loss
tag: Loss/regularization_loss



Loading model...Done!

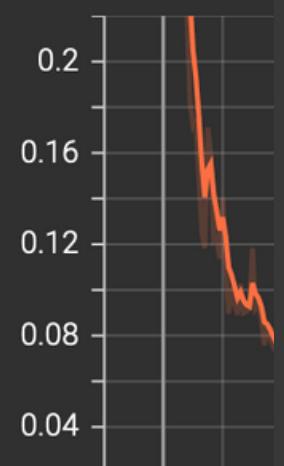
Fine-tuned Model



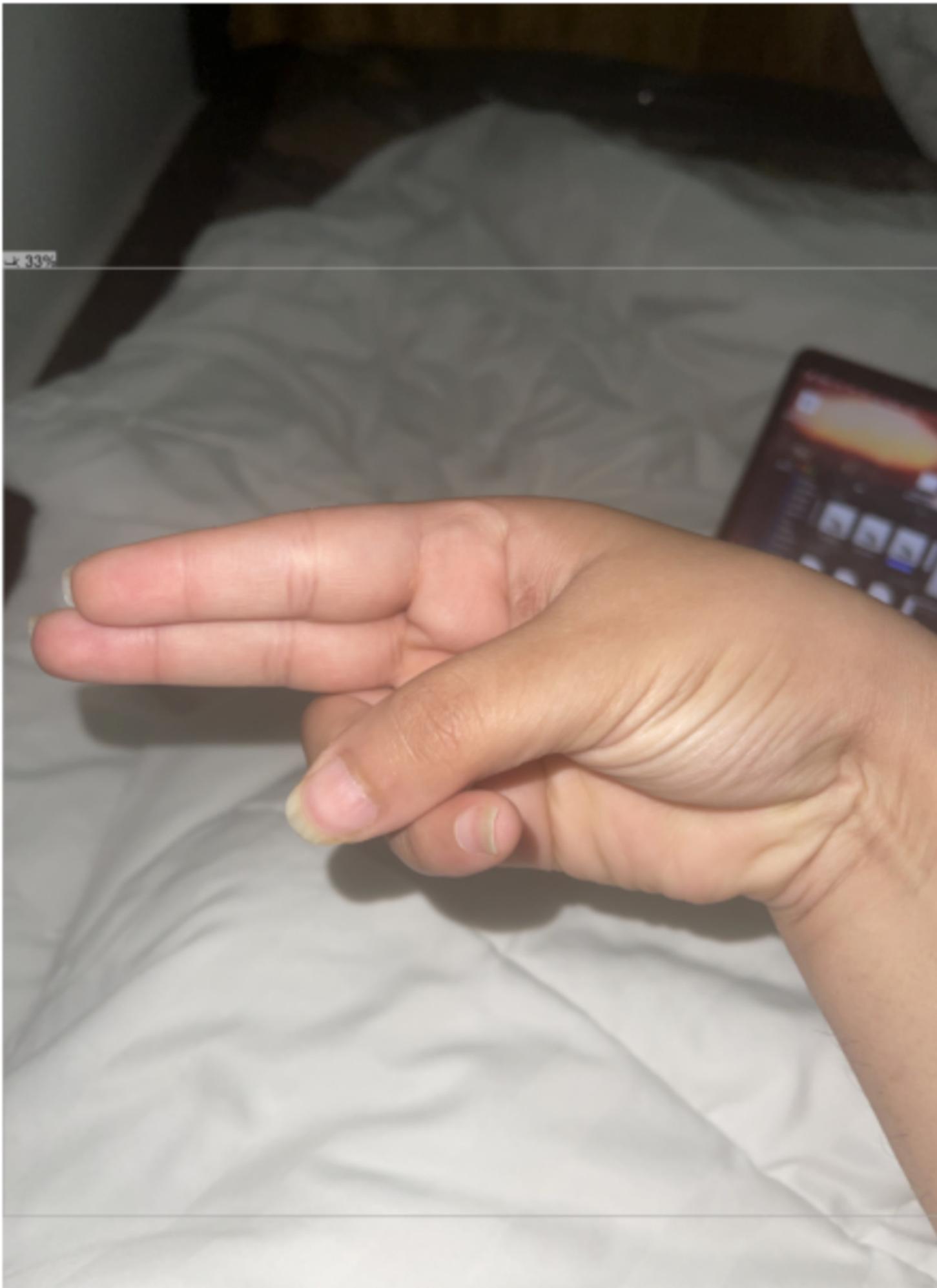
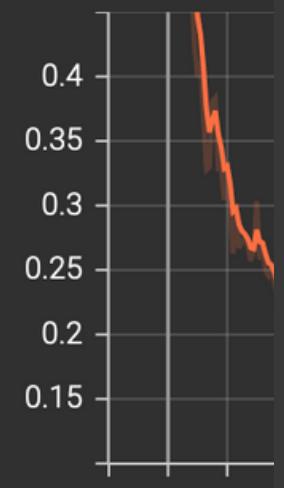
Loading model...Done!

Loss

Loss/classification
tag: Loss/classificatio



Loss/total_loss
tag: Loss/total_loss

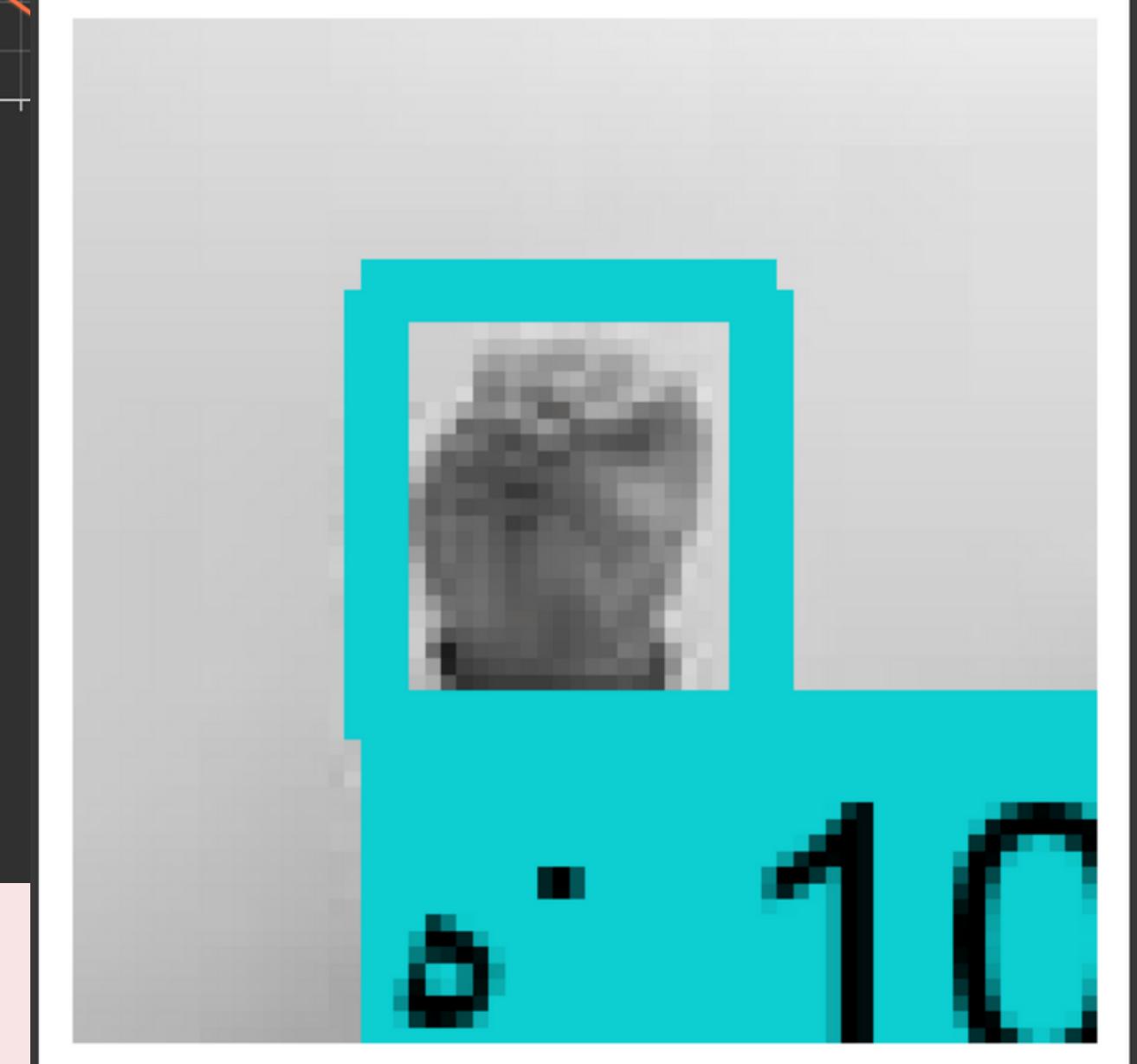


ation_loss
ization_loss

Loading model...Done!

4k 8k

Fine-tuned Model



CHALLENGES

Compatibility Issues

Python3 and Tensorflow

Tensorflow V1 and V2 APIs

Local environment was not compatible

Arabic Labels

Labellmg settings had to be adjusted to render
labels correctly in .xml files

+

Even after adjusting the encoding the model
visualization utility files had to be adjusted to
render arabic labels during detection

Long Training Time

4+ Hours

CHALLENGES

Compatibility Issues

Python3 and Tensorflow

Tensorflow V1 and V2 APIs

Local environment was not compatible

Arabic Labels

Labellmg settings had to be adjusted to render labels correctly in .xml files

+

Even after adjusting the encoding the model visualization utility files had to be adjusted to render arabic labels during detection

Long Training Time

4+ Hours

THINGS I'VE TRIED

Real-Time Detection

1. Workaround for google colab > **No time**
2. Recording a video and performing live detection on it > **Google colab failed to record a working video on different browsers and different MIME types**
3. Self-recorded a video and fed it to the model for detection > **It failed to handle the video and detect**

Extracting Labels for Interpretation

Tried a straightforward way of extracting the label > **Didn't work + there was no time left**

FUTURE WORK

- Train the model on images that have better resolution and overall quality
- To achieve real-time detection and deploy the model as a mobile application to reach a broader demographic.
- Train the model on video instead of images to expand its capabilities to interpret ASL more effectively.
- Train the model on words to replicate how deaf and mute people actually use the sign language.



Misk مisk
مؤسسة محمد بن سلمان
Mohammed Bin Salman
Foundation

THANK YOU FOR LISTENING!

Feel free to ask questions

Follow me on my social media:

LinkedIn + Twitter: @saraahawi