

INDIANA UNIVERSITY BLOOMINGTON

IMAGE SEGREGATOR

Project By:
Drishti Dhamejani
Rutvik Parvatneni
Sai Prasad Parsa
Sahithi Ancha

The Science of the future

Today's Session

TOPICS

Introduction

Dataset

Research Questions

Phases of Implementation

INTRODUCTION

- Motivation:
"Saving time and memory are amongst the most crucial problems of modern world".

Research Questions



IMAGES WITH
PEOPLE AND
OTHER IMAGES



CLASSIFICATION
OF IMAGES INTO
TEXT AND
MEMES



PICTURES OF
FRIENDS OR
FAMILY
(PRIORITIES
SPECIFIED)

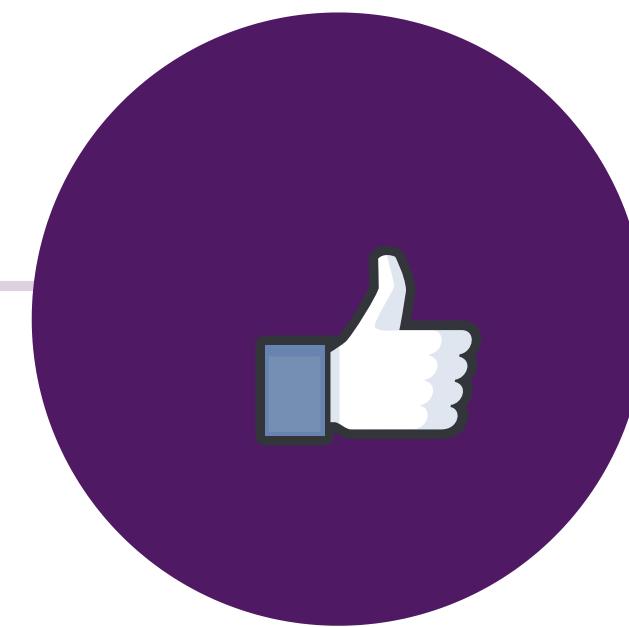
Dataset



Text



Humans



Memes

Previous Datasets

- We have previously used the following datasets ::
- Dataset- Version 1
 - Faces of People :
 - Georgia tech database :
 - Memes:
 - Python Image Scraper to download memes from Google.
 - Text:
 - Harvard dataset (IAMHandwriting database)

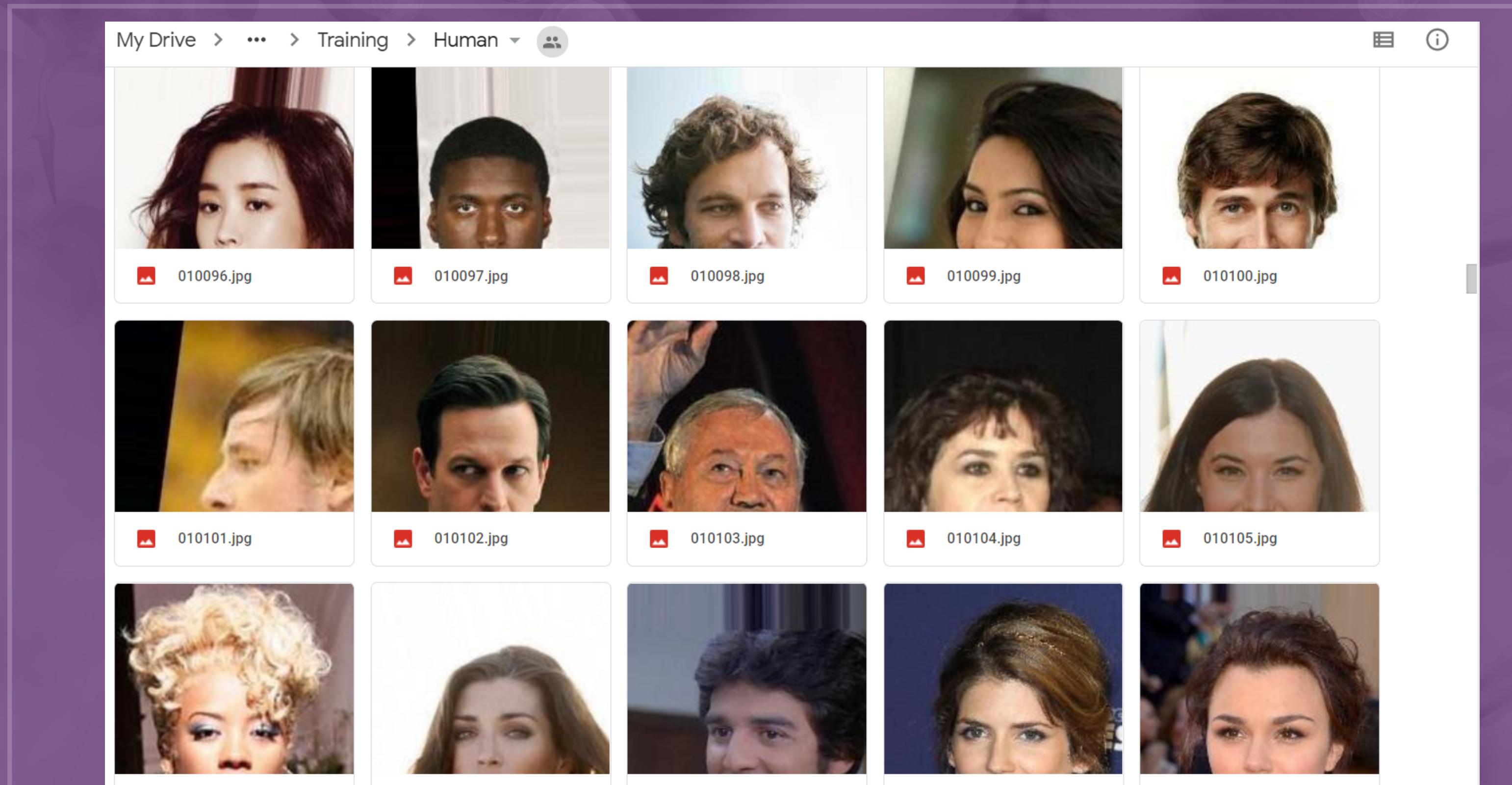
Previous Datasets

- We have previously used the following datasets ::
- Dataset- Version 2
 - Faces of People :
 - Celebrity dataset from kaggle
 - Memes:
 - Python Image Scraper to download memes from Instagram.

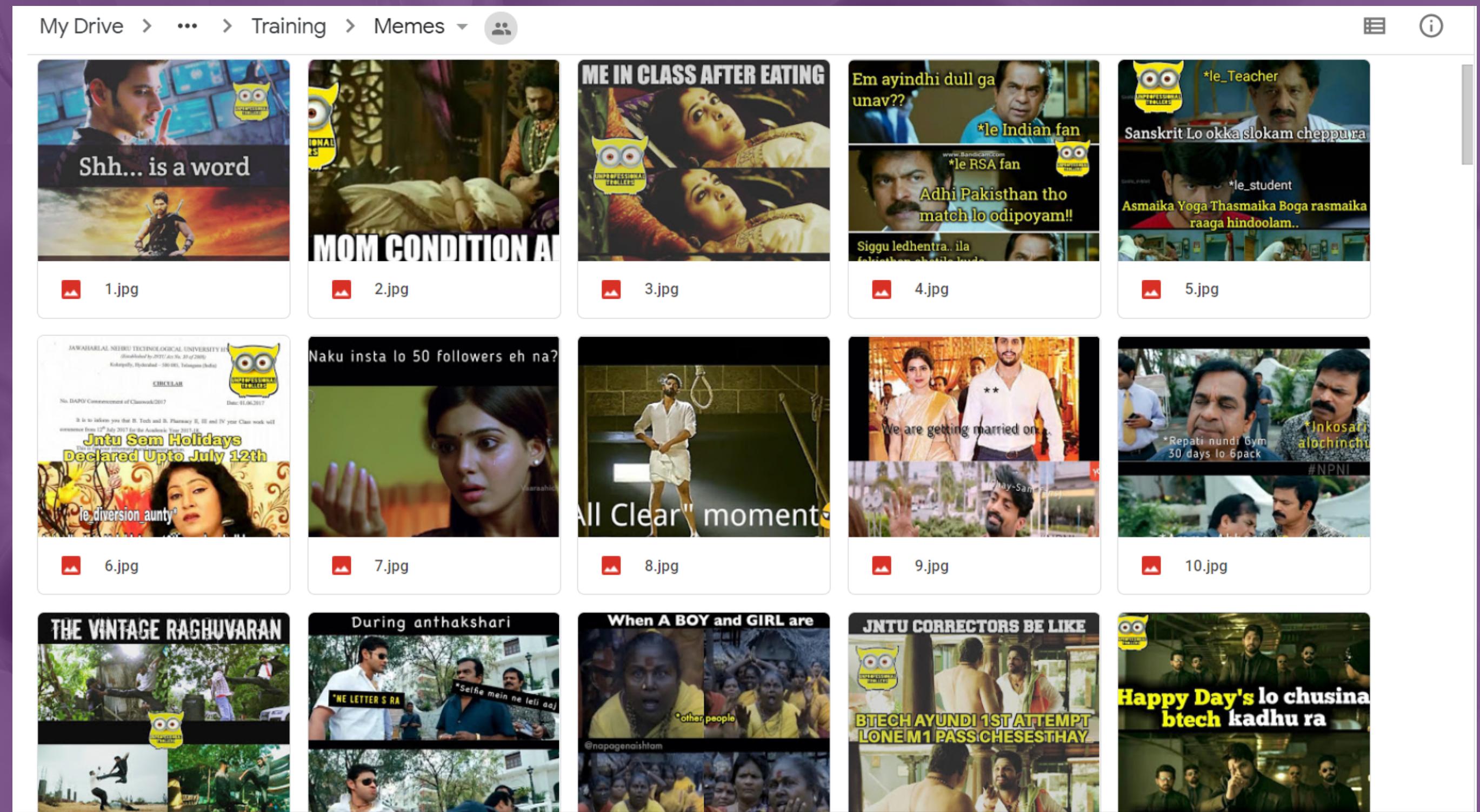
Final Datasets

- Dataset- Version 3
 - Faces of People :
 - Celebrity dataset from kaggle
 - Memes:
 - Python Image Scraper to download memes from Instagram.
 - Capdt
 - Meme_Raja
 - Meme_baasha
 - Fukkard
 - Text:
 - Used Image Scraper to download from multiple pages:
 - extreme_quotes, feelingsbyquotes, quotes_n_much_more, quotesbywehearit, sccopwhoopwordgasm , thegoodquote , wordporm.

Our Human- Dataset-3



Our Meme - Dataset-3



Our Text - Dataset-3

My Drive > ... > Training > Text

20046.jpg

Sometimes I wonder if anyone is happy to have me in their life.

20047.jpg

Happy 4th of JULY

20048.jpg

Sometimes quiet people really do have a lot to say... They're just being careful about who they open up to.
— Susan Gale

20049.jpg

Don't you think there is always something unspoken between two people?

20050.jpg

Trust the process. Your time is coming. Just do the work and the results will handle themselves.
- Tony Gaskins

20051.jpg

Late night conversations make you learn a lot about people.

20052.jpg

"Were we bad or just bad timing?"
(via -FeelingsByQuotes on IG)

20053.jpg

"And, she had this weird habit of **being herself** all the time;
that's why, **not everyone liked her.**"

20054.jpg

"And karma said you will fall in love with someone who doesn't love you, for not loving someone who did."

20055.jpg

Dear haters, I have so much more for you to be mad at. Just be patient!
@quotes_n_much_more

20056.jpg

A GIRL DOESN'T NEED TO BE YOUR MOTHER, DAUGHTER OR FRIEND TO BE RESPECTED.

HOW DO YOU FIND SELF-LOVE?
YOU DIG.
YOU ISOLATE AND
YOU ACHE FROM
BEING LONELY.
YOU HEAL.
YOU ACCEPT,
YOU LOOK IN THE
MIRROR AND
SEE GOD.

20058.jpg

Everyday may not be good, but there is something good in every day.
@quotes_n_much_more

20059.jpg

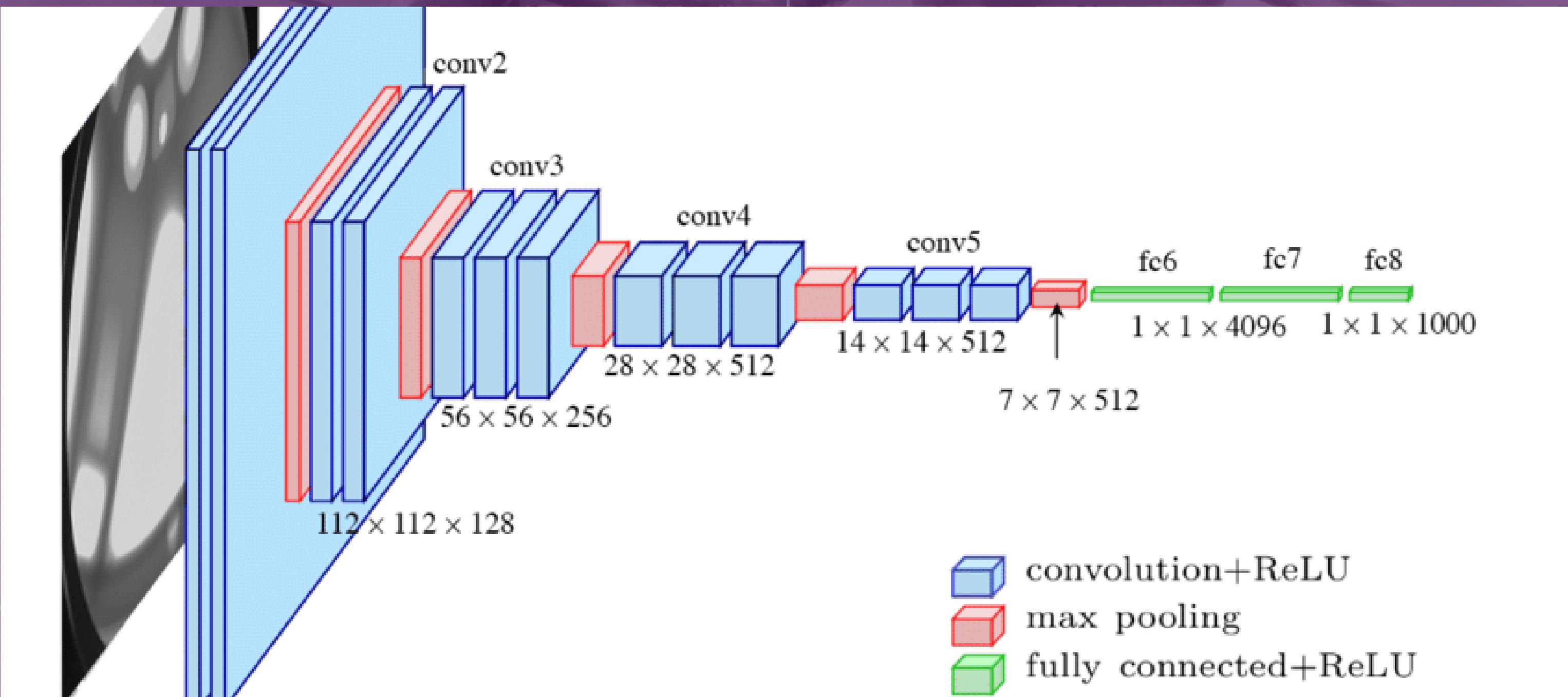
YES THE PAST CAN HURT, BUT THE WAY I SEE IT, YOU CAN EITHER RUN FROM IT, OR LEARN FROM IT.
WASNTAGONEKNOWN

YOUR EFFECT ON THOSE AROUND YOU GETS PASSED ALONG LIKE DOMINOES. WHAT CHAIN REACTION WILL YOU BEGIN TODAY?

ARCHITECTURES USED

FOR IMAGE CLASSIFICATION: VGG-16

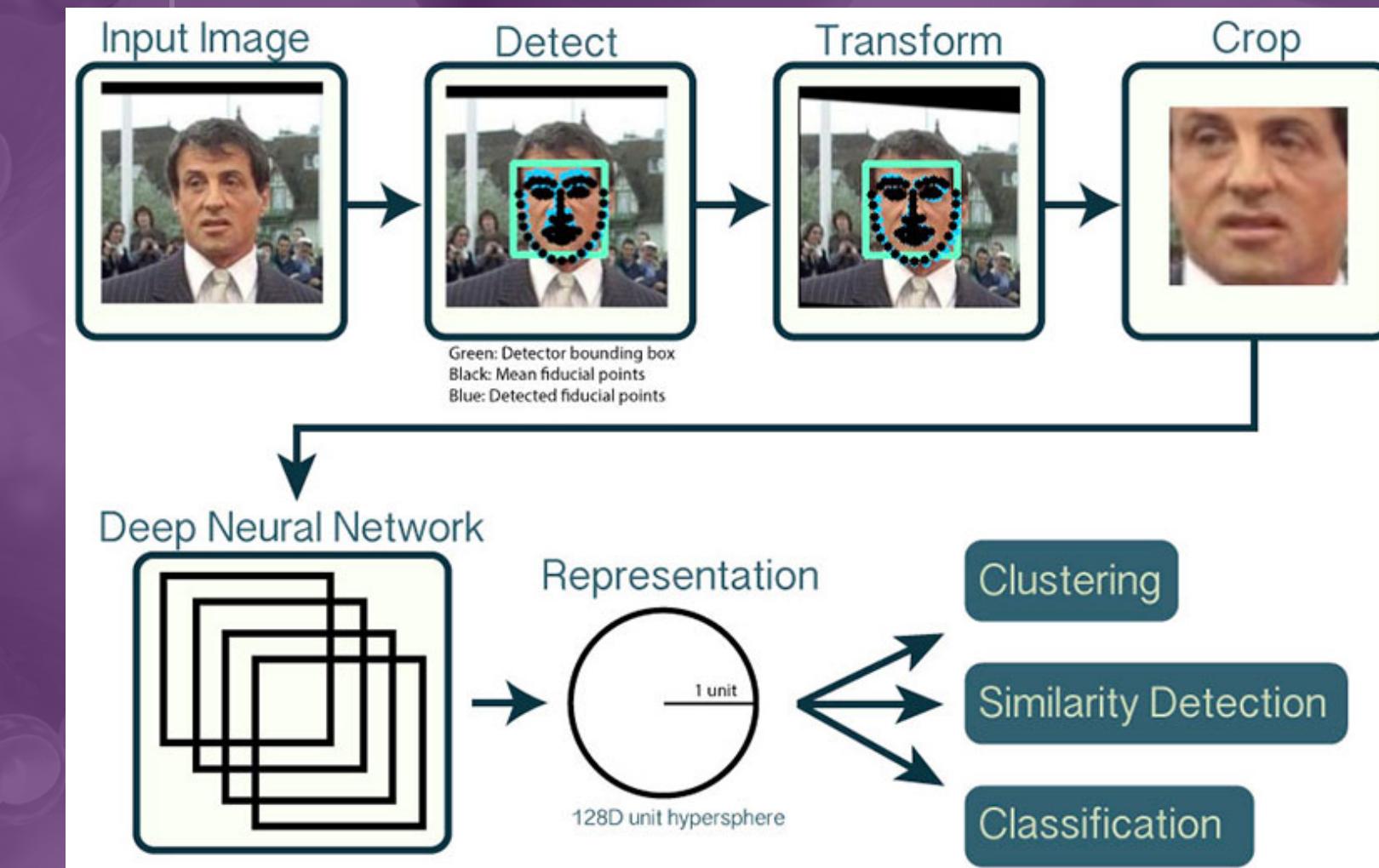
- Customized the network by removing 8 layers.
- Performed data augmentation (used only flip).



ARCHITECTURES USED

FOR FACE RECOGNITION: OPEN CV DEEP NEURAL NETWORK

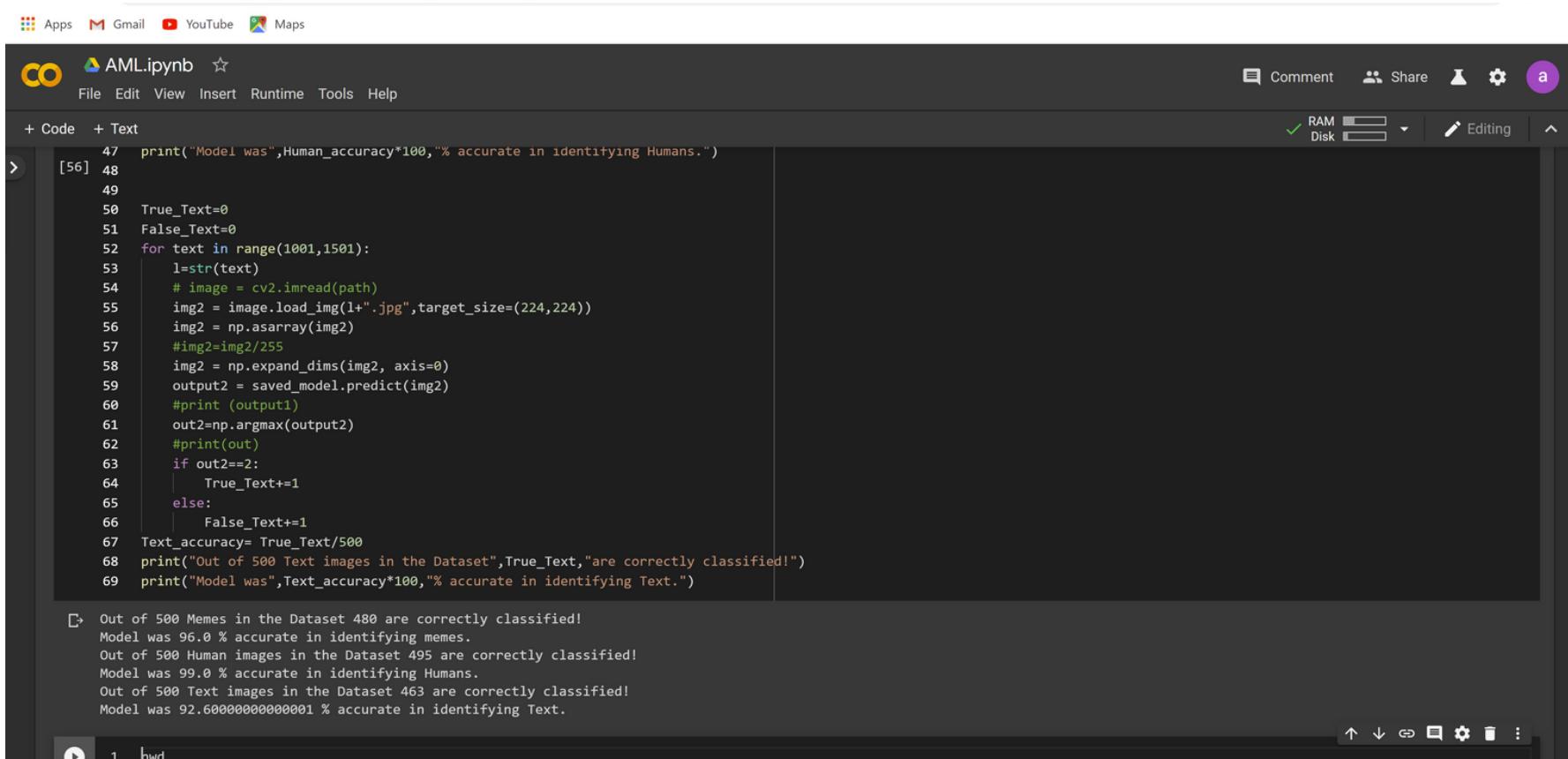
- Models used: Open Face project: A python and Torch implementation of 2015 CVPR publication.
- Used Deep Metric Learning.



ARCHITECTURES USED FOR FACE RECOGNITION: OPEN CV DEEP NEURAL NETWORK

- Steps :
 - Detect faces
 - Pre-trained Caffe deep learning model provided by OpenCV to detect faces
 - Compute 128-d face embeddings to quantify a face
 - A Torch deep learning model which produces the 128-D facial embeddings.
 - Train a Support Vector Machine (SVM) on top of the embeddings
 - Recognize faces in images

Human Recognition Accuracy



```
File Edit View Insert Runtime Tools Help
+ Code + Text
47 print("Model was",Human_accuracy*100,"% accurate in identifying Humans.")
48
49
50 True_Text=0
51 False_Text=0
52 for text in range(1001,1501):
53     l=str(text)
54     # image = cv2.imread(path)
55     img2 = image.load_img(l+"."+jpg,target_size=(224,224))
56     img2 = np.asarray(img2)
57     #img2=img2/255
58     img2 = np.expand_dims(img2, axis=0)
59     output2 = saved_model.predict(img2)
60     #print (output1)
61     out2=np.argmax(output2)
62     #print(out)
63     if out2==2:
64         True_Text+=1
65     else:
66         False_Text+=1
67 Text_accuracy= True_Text/500
68 print("Out of 500 Text images in the Dataset",True_Text,"are correctly classified!")
69 print("Model was",Text_accuracy*100,"% accurate in identifying Text.")

Out of 500 Memes in the Dataset 480 are correctly classified.
Model was 96.0 % accurate in identifying memes.
Out of 500 Human images in the Dataset 495 are correctly classified!
Model was 99.0 % accurate in identifying Humans.
Out of 500 Text images in the Dataset 463 are correctly classified!
Model was 92.600000000001 % accurate in identifying Text.
```

MODEL SUMMARY

ACCURACY :: 99%
images- 500

495 of 500 images were correctly classified.

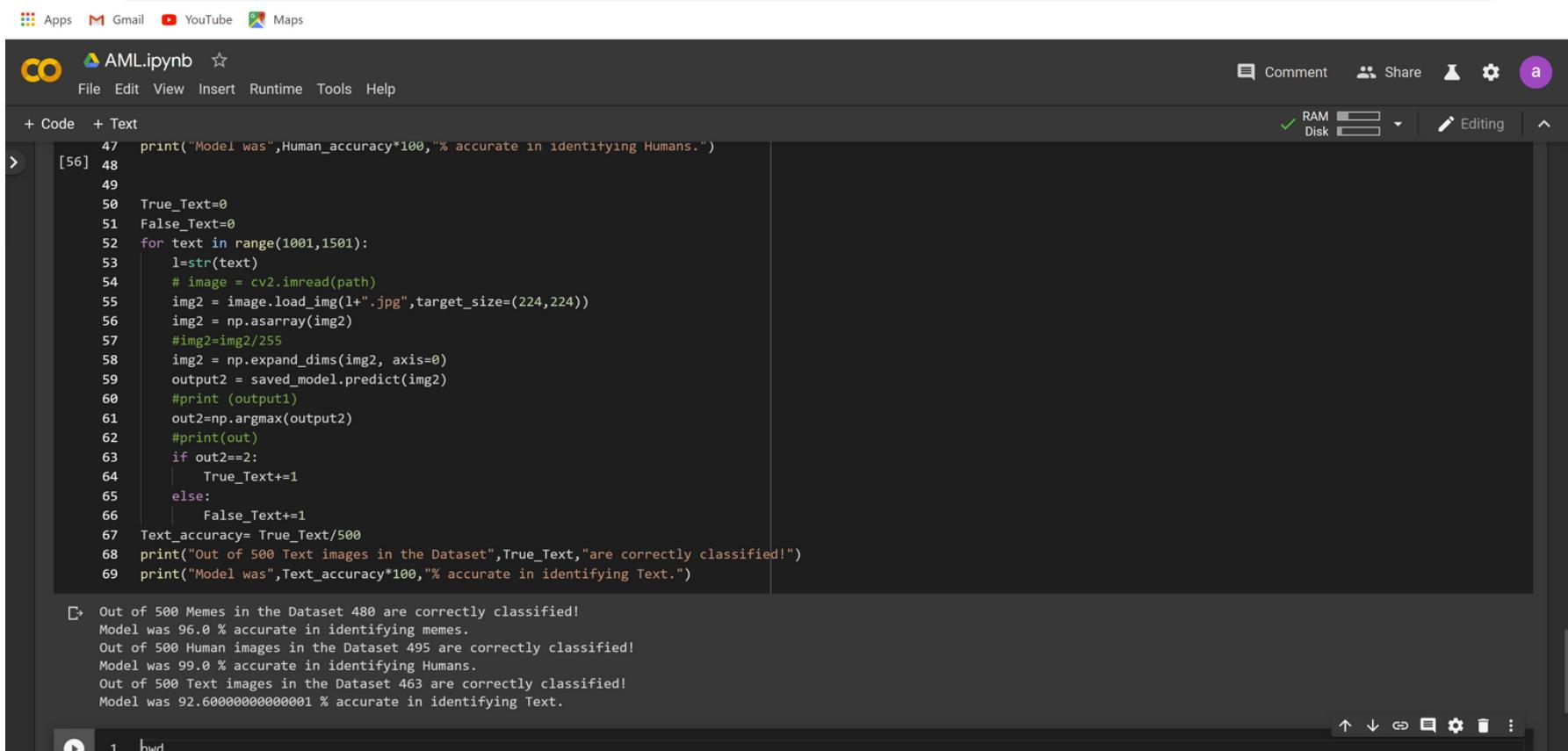
ACCURACY :: 86.67%
images- 15
(images with our faces)

```
Model was 86.66666666666667 % accurate in identifying Humans.

cd ..
```

13 of 15 images were correctly classified.

Meme Recognition Accuracy



A screenshot of a Jupyter Notebook cell titled "AML.ipynb". The code in the cell is as follows:

```
47 print("Model was",Human_accuracy*100,"% accurate in identifying Humans.")
[56] 48
49
50 True_Text=0
51 False_Text=0
52 for text in range(1001,1501):
53     l=str(text)
54     # image = cv2.imread(path)
55     img2 = image.load_img(l+"."+jpg",target_size=(224,224))
56     img2 = np.asarray(img2)
57     #img2=img2/255
58     img2 = np.expand_dims(img2, axis=0)
59     output2 = saved_model.predict(img2)
60     #print (output1)
61     out2=np.argmax(output2)
62     #print(out)
63     if out2==2:
64         True_Text+=1
65     else:
66         False_Text+=1
67 Text_accuracy= True_Text/500
68 print("Out of 500 Text images in the Dataset",True_Text,"are correctly classified!")
69 print("Model was",Text_accuracy*100,"% accurate in identifying Text.")
```

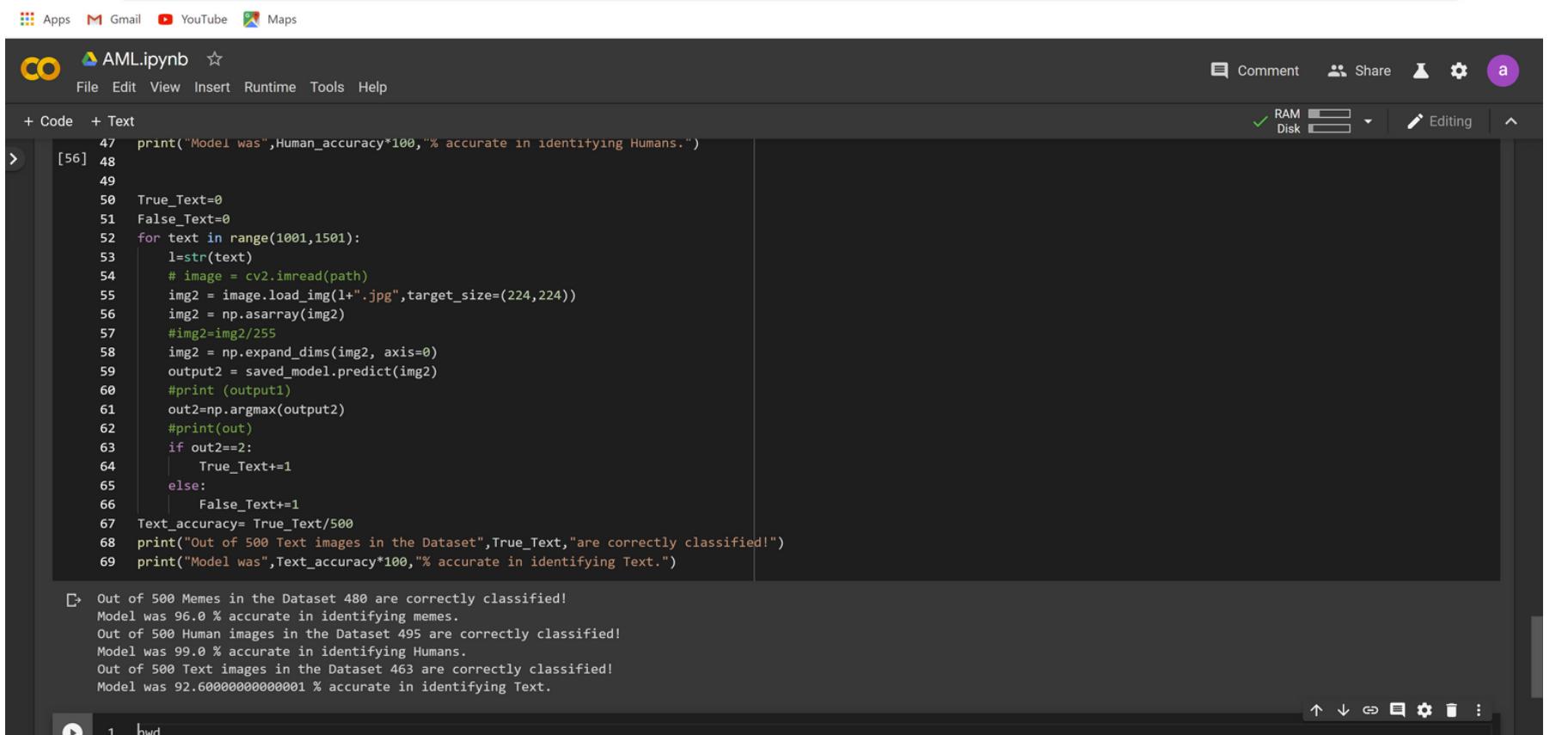
The cell output shows the results of the model's performance:

```
Out of 500 Memes in the Dataset 480 are correctly classified!
Model was 96.0 % accurate in identifying memes.
Out of 500 Human images in the Dataset 495 are correctly classified!
Model was 99.0 % accurate in identifying Humans.
Out of 500 Text images in the Dataset 463 are correctly classified!
Model was 92.600000000001 % accurate in identifying Text.
```

MODEL SUMMARY
ACCURACY :: 96%
images- 500

480 images of 500 were
correctly classified.

Text Recognition Accuracy



A screenshot of a Jupyter Notebook cell titled "AML.ipynb". The code in the cell is as follows:

```
47 print("Model was",Human_accuracy*100,"% accurate in identifying Humans.")
[56] 48
49
50 True_Text=0
51 False_Text=0
52 for text in range(1001,1501):
53     l=str(text)
54     # image = cv2.imread(path)
55     img2 = image.load_img(l+"."+jpg",target_size=(224,224))
56     img2 = np.asarray(img2)
57     #img2=img2/255
58     img2 = np.expand_dims(img2, axis=0)
59     output2 = saved_model.predict(img2)
60     #print (output1)
61     out2=np.argmax(output2)
62     #print(out)
63     if out2==2:
64         True_Text+=1
65     else:
66         False_Text+=1
67 Text_accuracy= True_Text/500
68 print("Out of 500 Text images in the Dataset",True_Text,"are correctly classified!")
69 print("Model was",Text_accuracy*100,"% accurate in identifying Text.")
```

The cell output shows the results of the model's performance:

```
Out of 500 Memes in the Dataset 480 are correctly classified!
Model was 96.0 % accurate in identifying memes.
Out of 500 Human images in the Dataset 495 are correctly classified!
Model was 99.0 % accurate in identifying Humans.
Out of 500 Text images in the Dataset 463 are correctly classified!
Model was 92.600000000001 % accurate in identifying Text.
```

MODEL SUMMARY
ACCURACY :: 92.6%
images- 500

463 images of 500 were correctly classified.

Overall Accuracy

MODEL SUMMARY

Overall - ACCURACY :: 95.8%
images- 1500

1438 of 1500 images were
correctly classified .

Face Recognition Accuracy

```
No.of images in Priority 8
No.of iamges in others 5

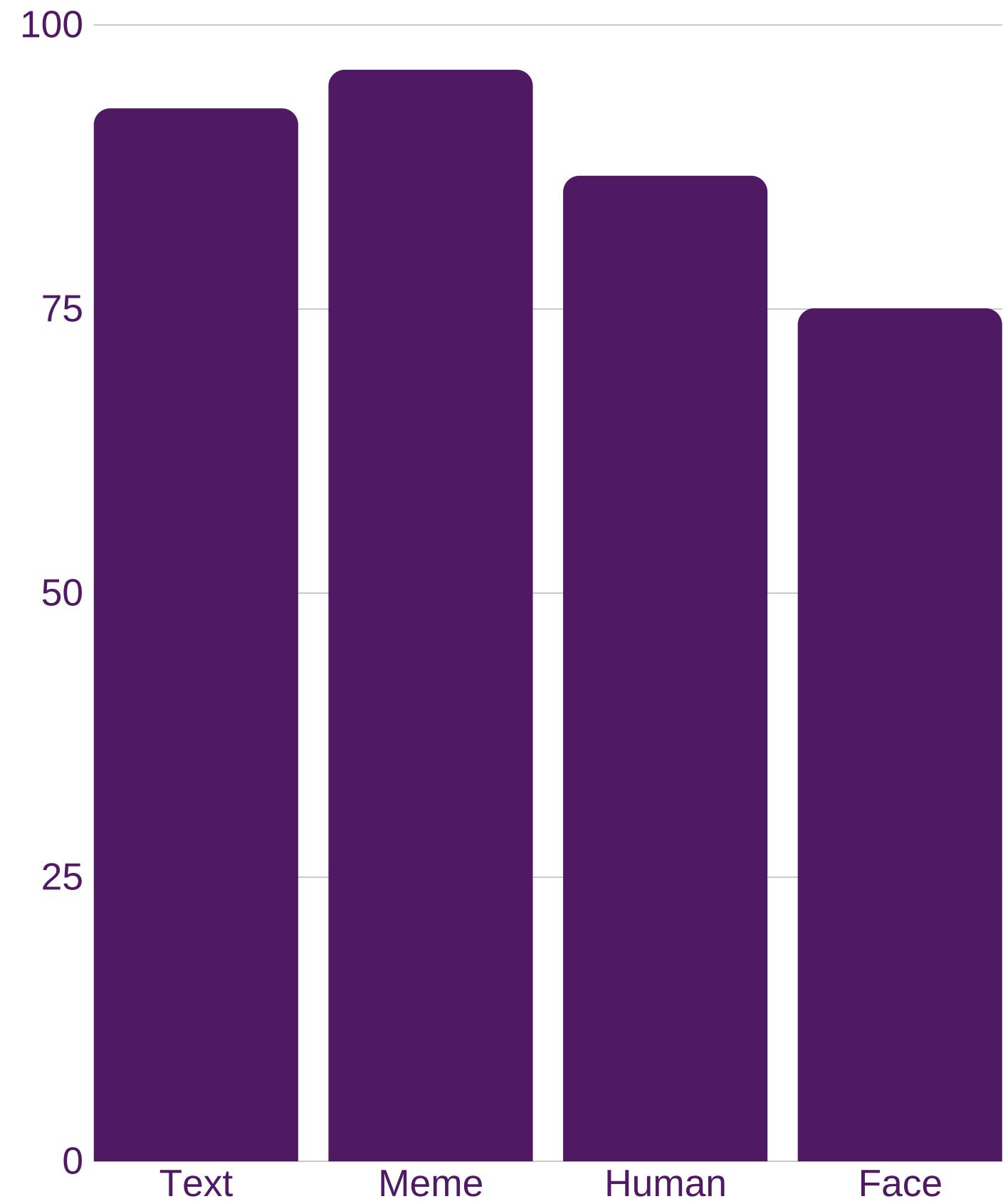
[26]: 1 Priority_images_in_test1=6 # The number of priority images in the test set
      2 Classified_as_priority= priorityc #no.of images the model identified as priority
      3
      4 Accuracy= Priority_images_in_test1/Classified_as_priority
      5 print (Accuracy*100,'percent')# Though all the priority images are correctly classified there are 2 images which are wrongly classified

In [26]: 75.0 percent
```

MODEL SUMMARY
ACCURACY :: 75%

This model detects all the priority images correctly with a few false detections.

True Positive : 6 images
False Positive: 2 images



Accuracy

Q&A

