BUE
The British University in Egypt
الجامعة البريطانية في مصر

On AI-Driven Privacy Protection: A Study on AI-based Solutions using Face Blurring Techniques

Done By: Sara Amjad Sabih Anabtawi 212071
Supervised by: Doctor Amr Ghoneim

Table of content

1. Introduction
2. Problem Statement
3. Scope and Objectives
4. State of the art
5. Proposed Solution
6. Targeted users
7. Datasets
8. Pipeline
9. Preprocessing
10. Blurring Techniques
11. Testing and Evaluation
12. Results
13. Challenges and Limitations
14. Conclusion

01

Introduction

In the digital age, the vast circulation of images online raises significant privacy concerns.

The project focuses on developing an efficient AI system for robust facial detection and blurring, making faces unrecognizable to advanced AI models like YOLO and MobileNet. Multiple blurring techniques may be applied to ensure the individual privacy is protected by blurring faces of non- consenting individuals.

Despite the blurring, image quality remains intact, maintaining integrity while enhancing digital privacy.

This practical tool will empower people to control their online identities, reduce unauthorized face visibility, and increase community safety. The project's outcomes will advance AI-driven privacy protection strategies, setting new standards in this critical field.

02

Problem Statement

- Many platforms allow the publication of images without consent, increasing risks such as :
  - Identity theft
  - Cyberbullying
  - Reputational damage

03

Scope and Objective

- Scope:
  - Image Input (as various types)
  - Multiple Face Detection and Recognition
  - Selective Face Blurring
  - Preserve the quality

- Objective:

Privacy & Utility Preservation

04

State of the art

Paper	Year	Approach	Accuracy
2	2018	Haar Cascade R-CNN	83.8% 89.6%
3	2016	Haar classifier (Detection) PCA (Recognition)	98.18% 93.33%
4	2015	VGG-16 and VGG-19	98.95% 97.3%
5	2019	GoogLeNet	88%
1	2024	CNN	99.67%

05

Proposed Solution

Models For Face Detection :

- Yolov5
- Yolov8

Models For Face Recognition (Selective Blurring) :

- MobileNet
- Yolov5
- Yolov8

Blurring techniques :

- Gaussian Blur
- Median Blur
- Box Filter Blur
- Mean Shift Filtering
- Radial Blur
- Motion Blur
- GrabCut/Guassian Blur

Models For Face Recognition(on blurred faces):

- MobileNet
- Yolov5
- Yolov8

06

Targeted Users

- Businesses: Protecting Customer Privacy in Marketing Materials, and other Visual Content.
- Individuals: Sharing Personal Photos on Social Media without Exposing the Identities of others Captured in the background.
- Journalists: Blurring Faces(sources) in Sensitive Interviews or Investigations.

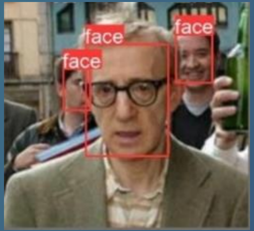
07

Datasets

LFW(Labeled Faces in the Wild) **Dataset**

This Dataset includes annotated images with bounding boxes around faces, making it suitable for training YOLO models for face detection.

- Images: 13,231 images of faces, each labeled with the person's name.
- Bounding Boxes: Each image is annotated with bounding boxes that indicate the position of the face, which is essential for training YOLO models.
- Variability: The images were taken in uncontrolled environments with various lighting conditions, poses, and backgrounds, providing a robust dataset for face detection.



Avengers Dataset

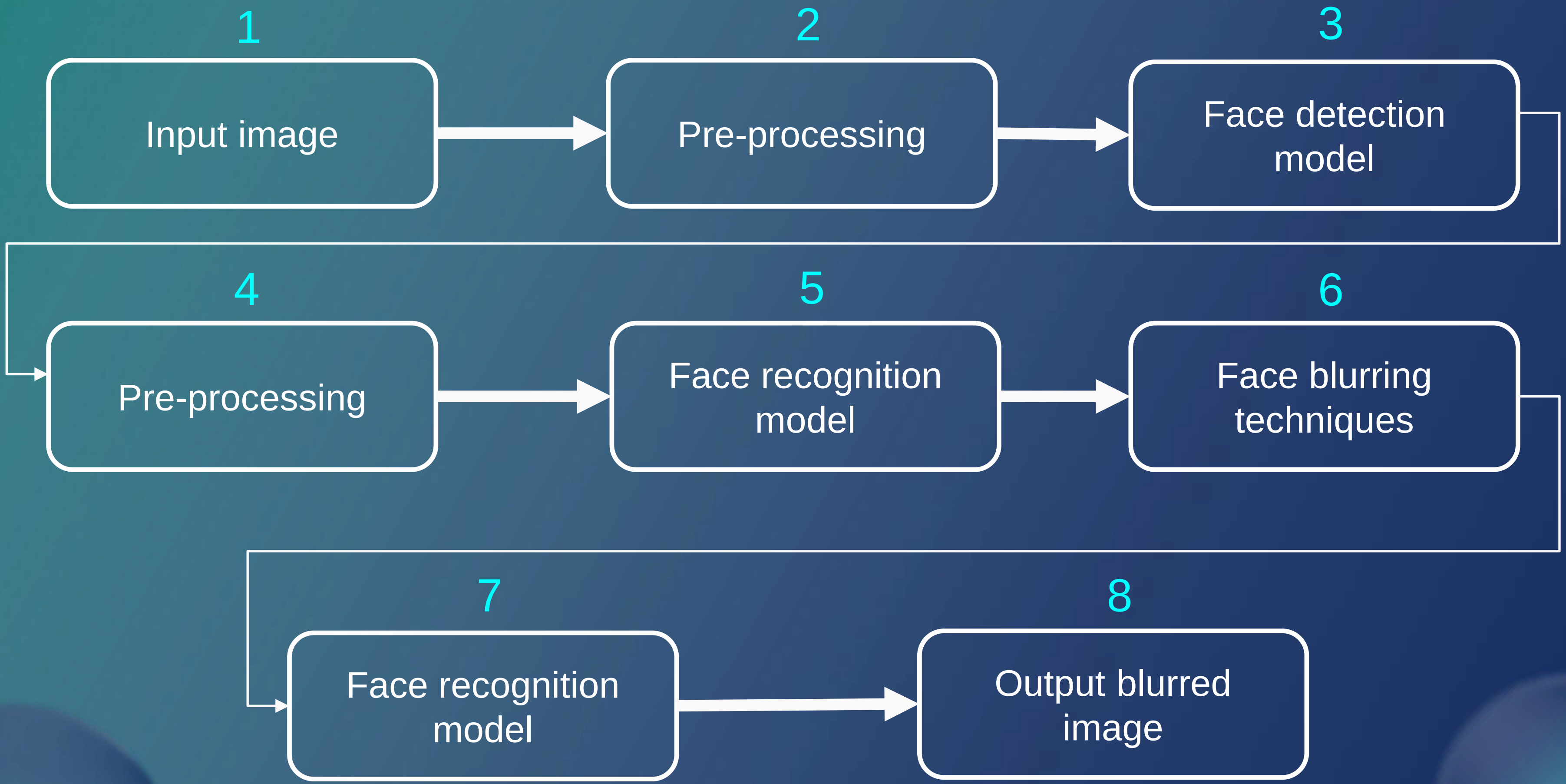
The Avengers dataset contains images of six characters from the Avengers movie series. This dataset was labelled but is no annotated with bounding boxes and wasn't divided.

- Images: 781 images of cropped faces
- Classes:

- Tony Stark (Robert Downey Jr.)
- Steve Rogers (Chris Evans)
- Natasha Romanoff (Scarlett Johansson)
- Bruce Banner (Mark Ruffalo)
- Clint Barton (Jeremy Renner)
- Thor (Chris Hemsworth)

08

Generic Pipeline



1 Input image

2 Pre-processing

3 Face detection model

4 Pre-processing

5 Face recognition model

6 Face blurring techniques

7 Face recognition model

8 Output blurred image

09

Pre-Processing

Face Recognition Model Yolov5/Yolov8 Preprocessing

- ❑ load and convert images to RGB format.

- ❑ Load the Utilized pretrained YOLOv5/Yolov8 model, that was fine-tuned on the LFW dataset for face detection.

- ❑ Detecting Faces and Generating Bounding Boxes
- ❑ Save the annotations of each image
- ❑ Draw bounding box on each image
- ❑ Convert the annotations to yolo format
- ❑ divide a dataset of images and their corresponding bounding box labels into training and validation sets using an 80:30 ratio
- ❑ Move files so that the images are in a folder of 2 Folders (train and val) and in each folder it has 6 sub folders(6 classes) and in each of these folders there will be two sub folders (image and label).
- ❑ Create Yaml File

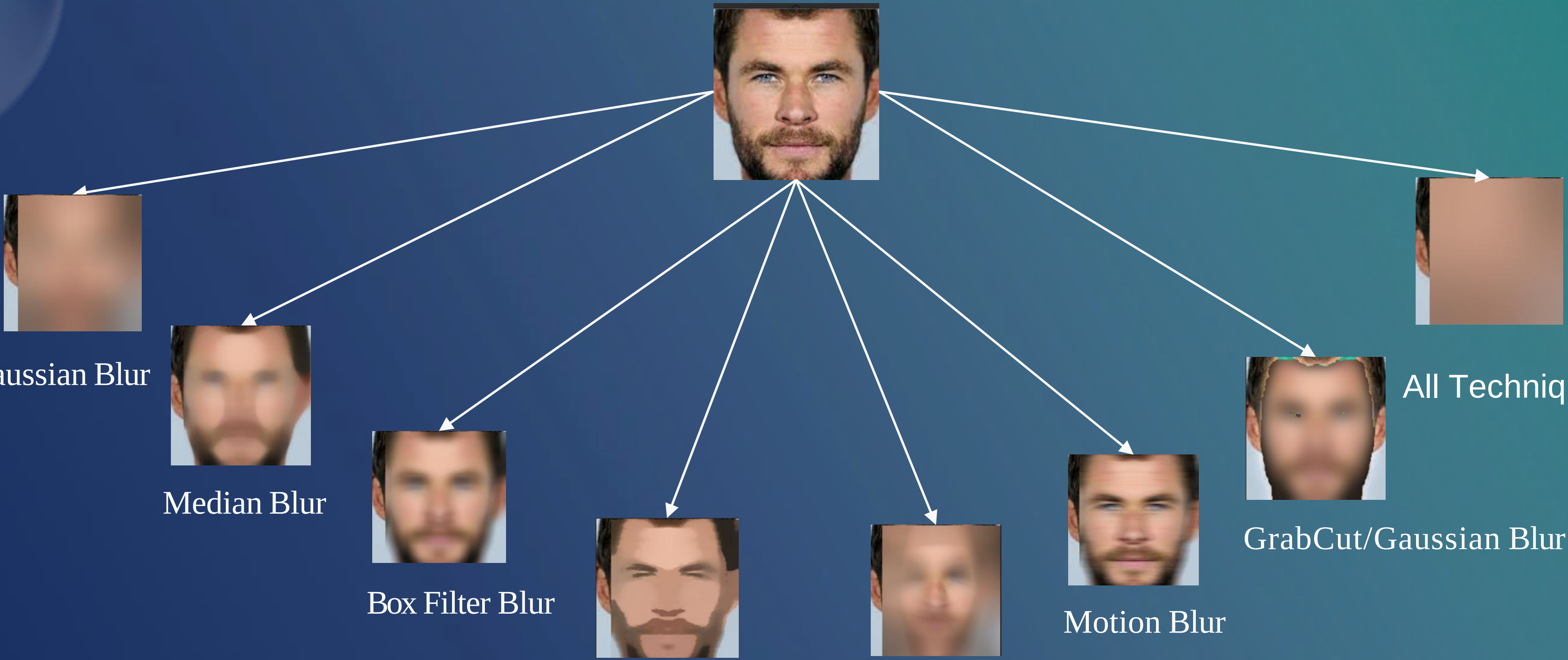
Face Recognition Model MobileNet Preprocessing

- Load and convert images to RGB format.
- Resize images to 224x224 Pixels
- Normalize images
- Encode Classes using a OneHotEncoder
- Split Dataset into Training and testing using a 70:30 ratio
- Adding New Top Layers such as global average pooling, a few dense layers with ReLU activation, batch normalization, dropout for regularization, and a SoftMax layer for classification.

10

Blurring Techniques Sample

Original Image



Gaussian Blur

Median Blur

Box Filter Blur

Mean Shift Filtering

Radial Blur

Motion Blur

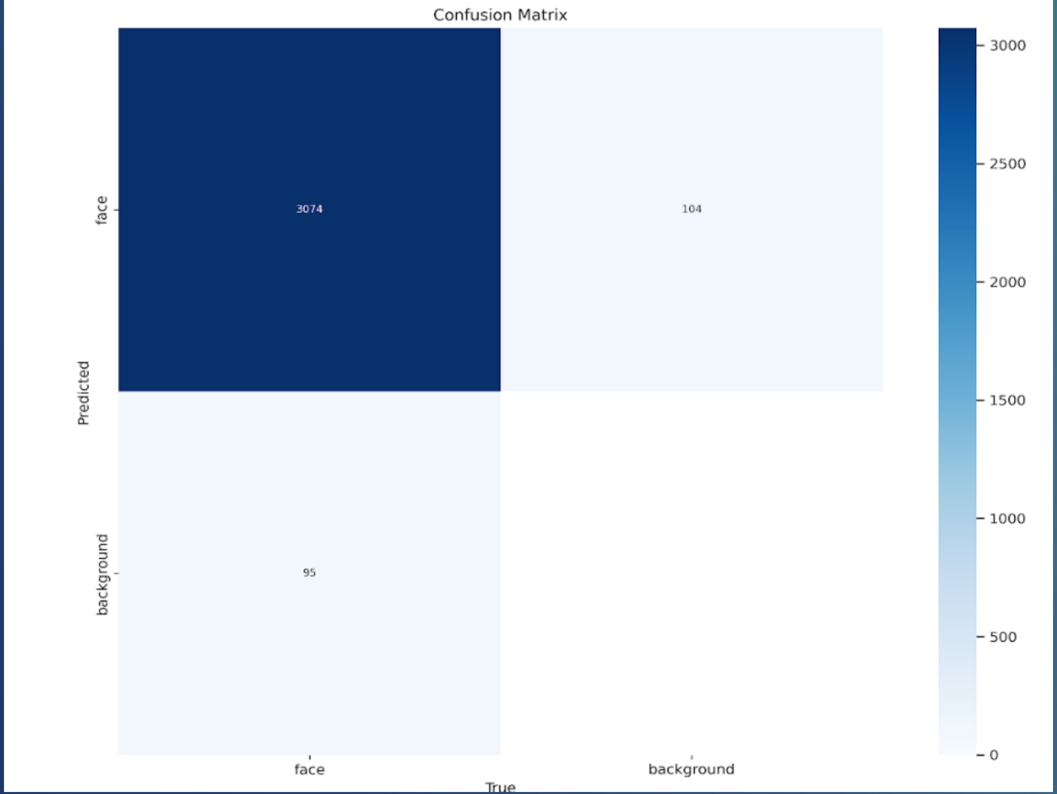
GrabCut/Gaussian Blur

All Techniques

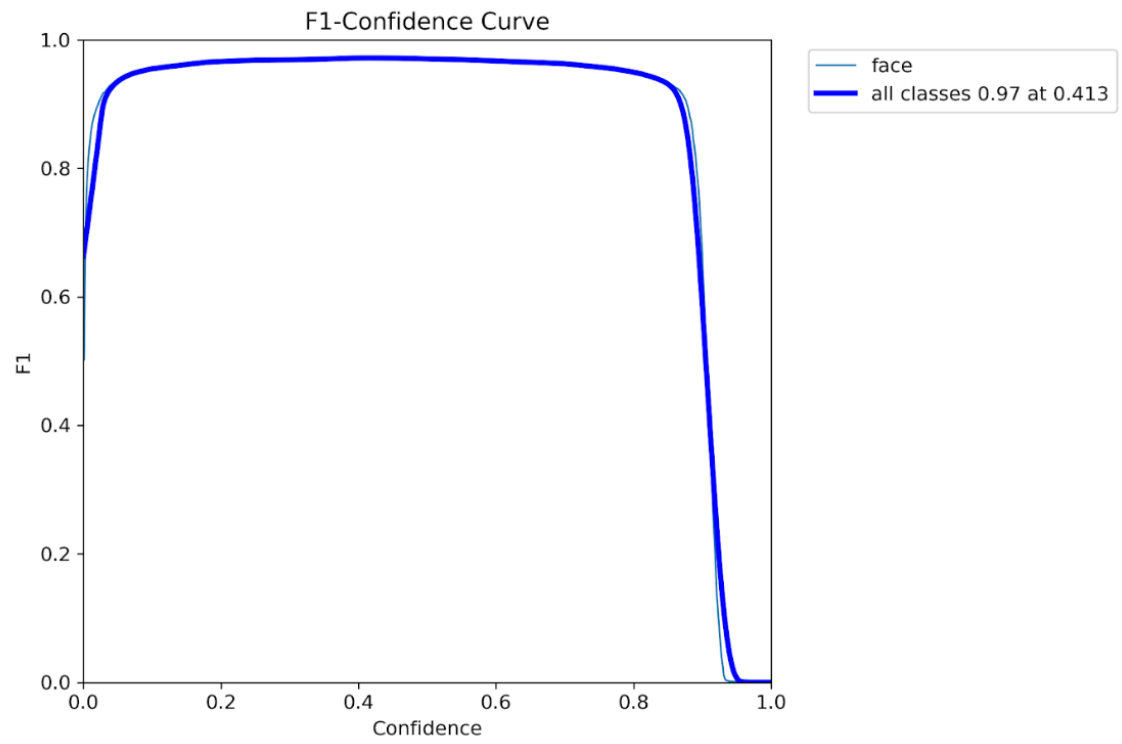
11

Testing and Evaluation

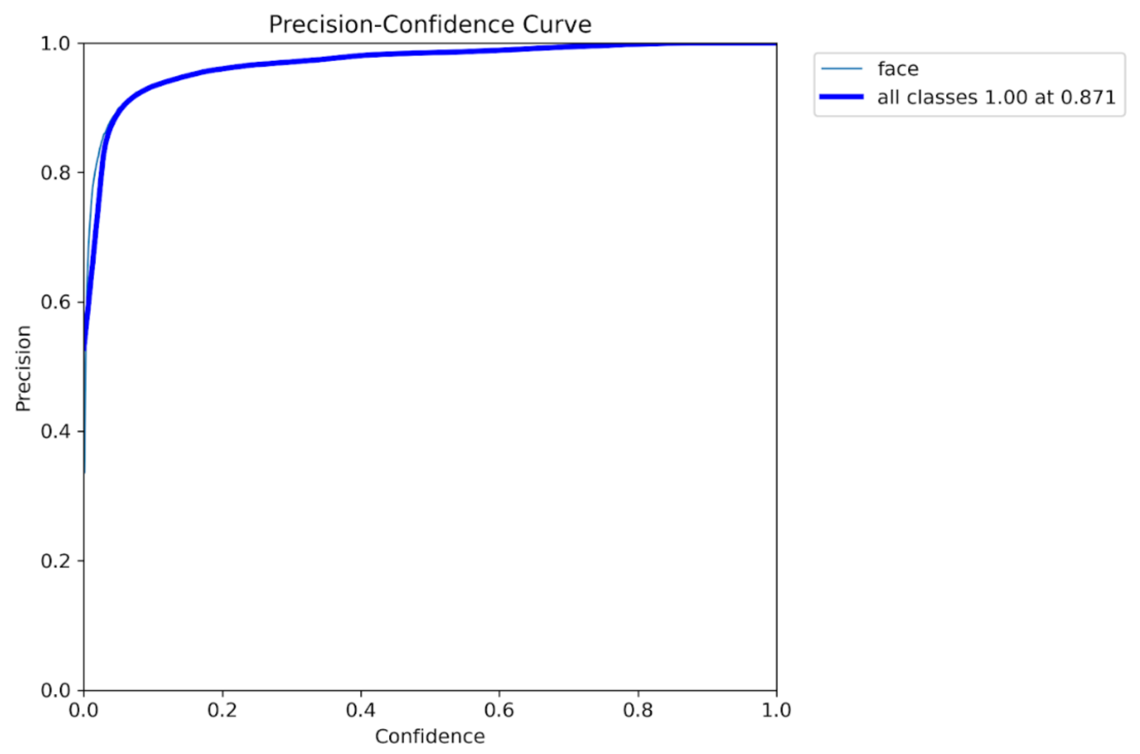
1. YOLOV8 as a Face Detection Model(15 epochs) :



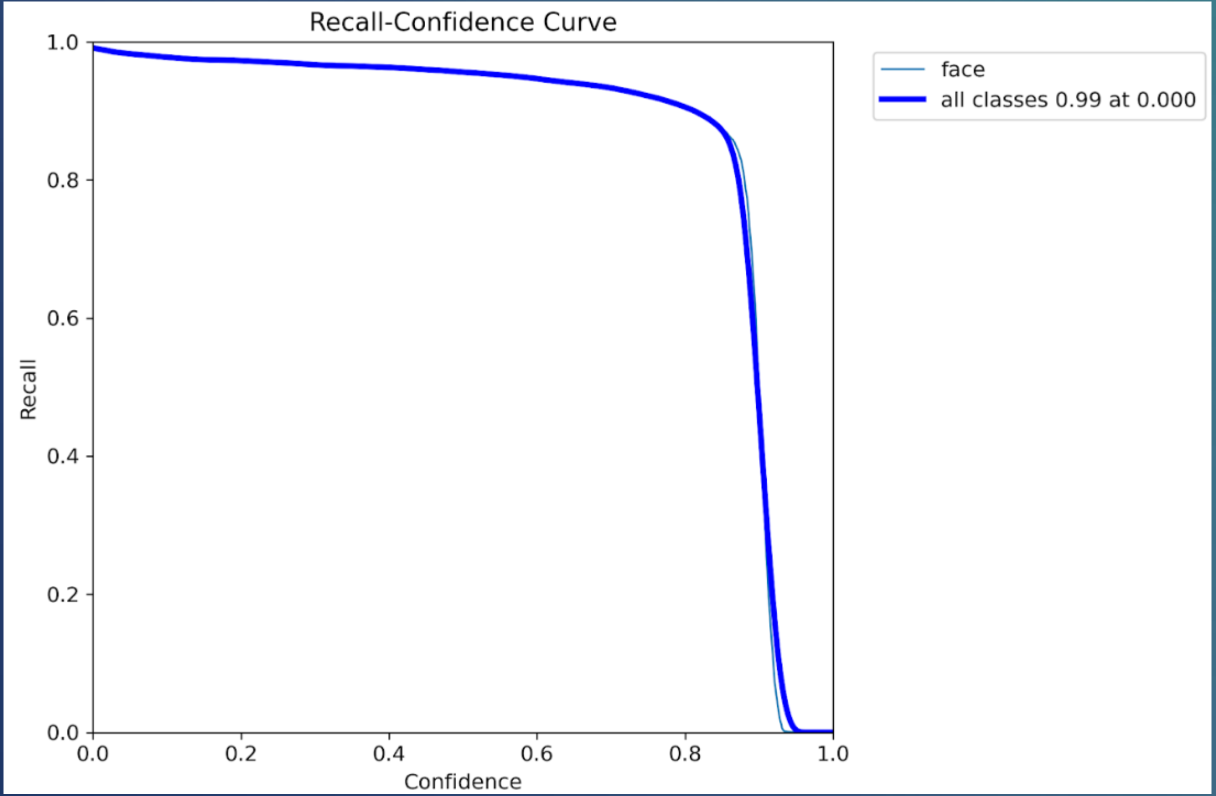
1. YOLOV8 as a Face Detection Model(15 epochs) :



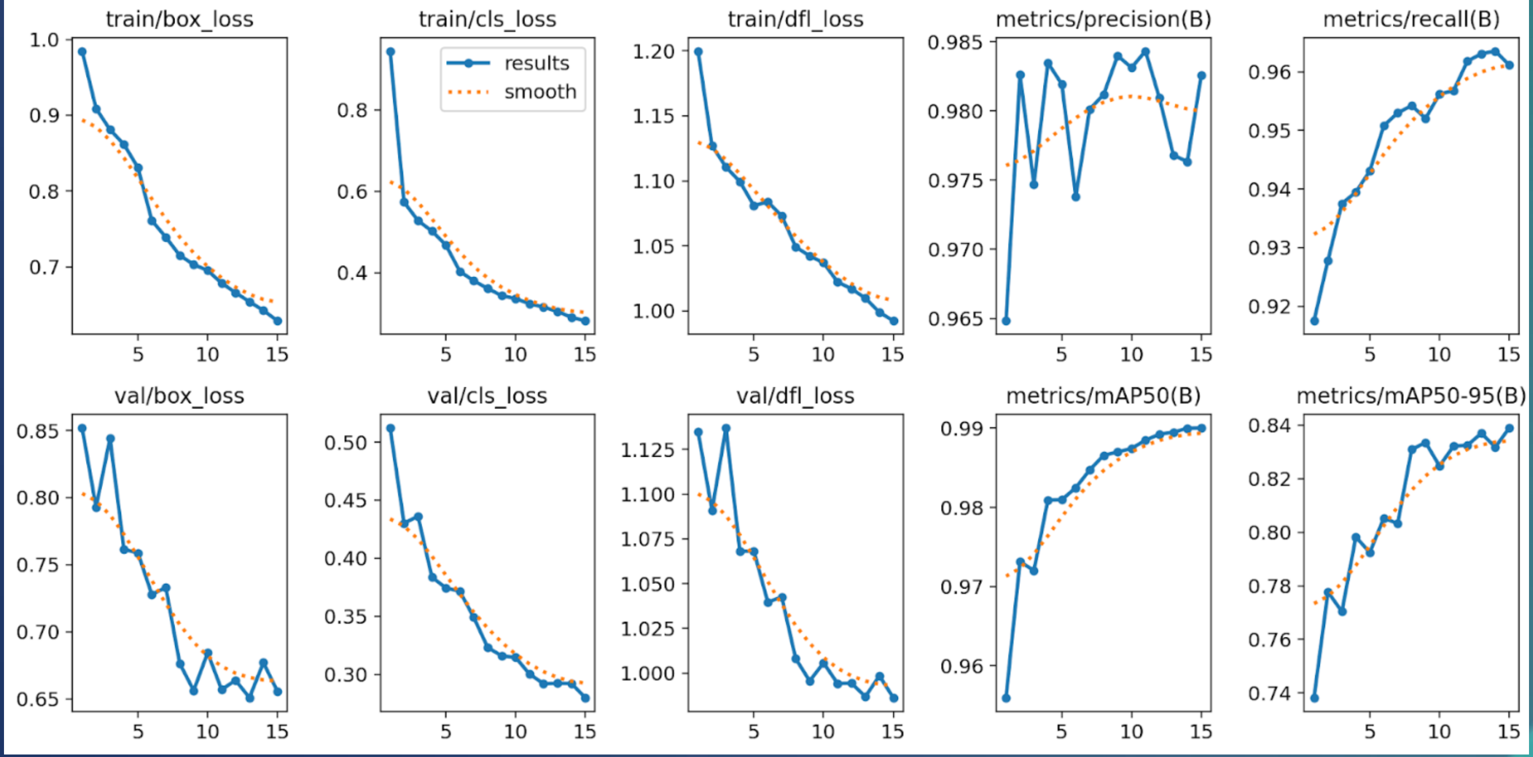
1. YOLOV8 as a Face Detection Model(15 epochs) :



1. YOLOV8 as a Face Detection Model(15 epochs) :

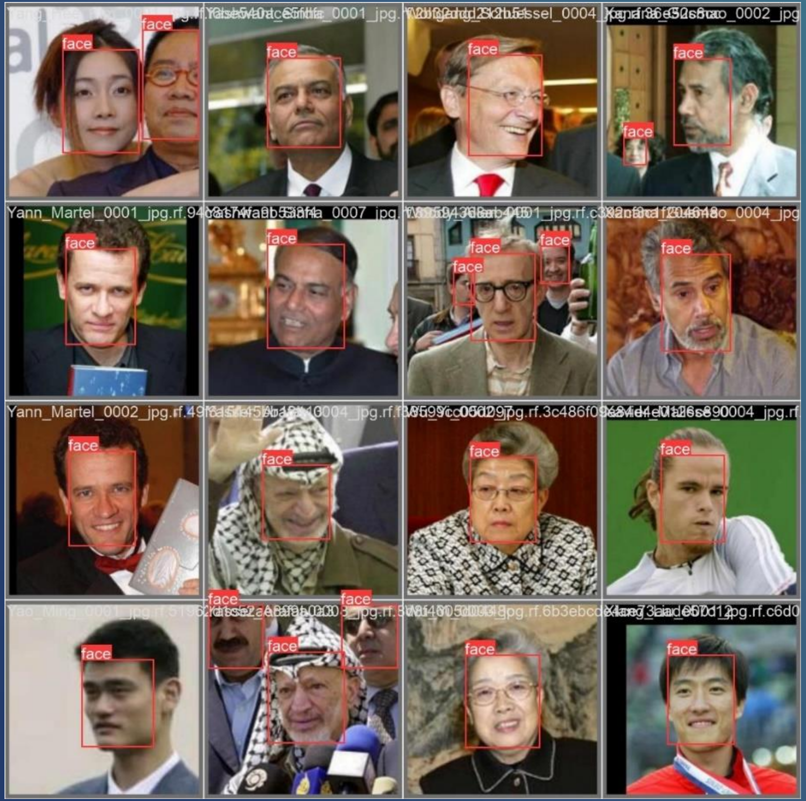


1. YOLOV8 as a Face Detection Model(15 epochs) :



1. YOLOV8 as a Face Detection Model(15 epochs) :

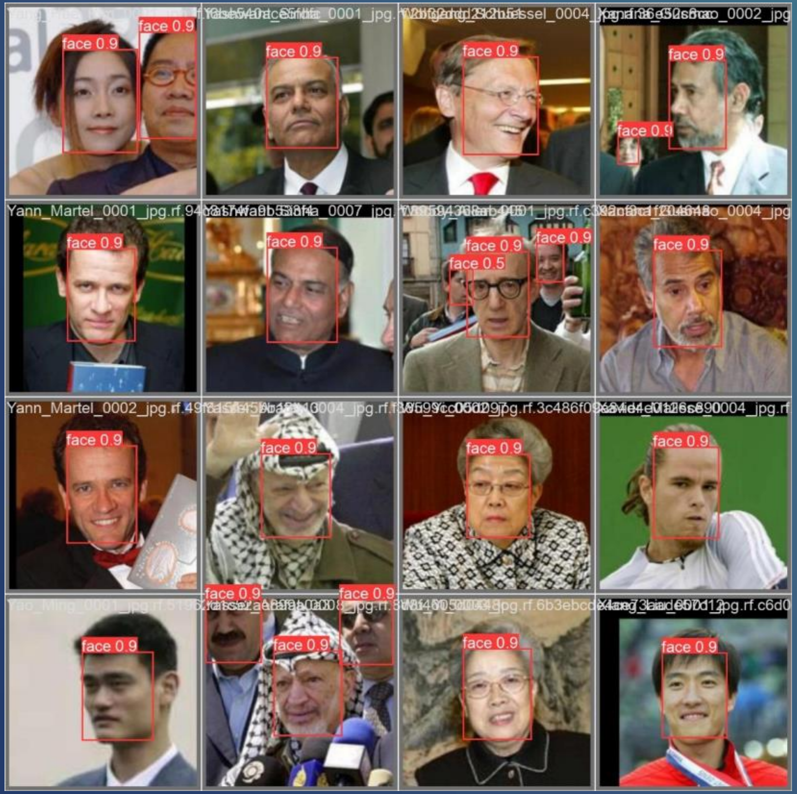
Validation Sample Batch 1 Labels



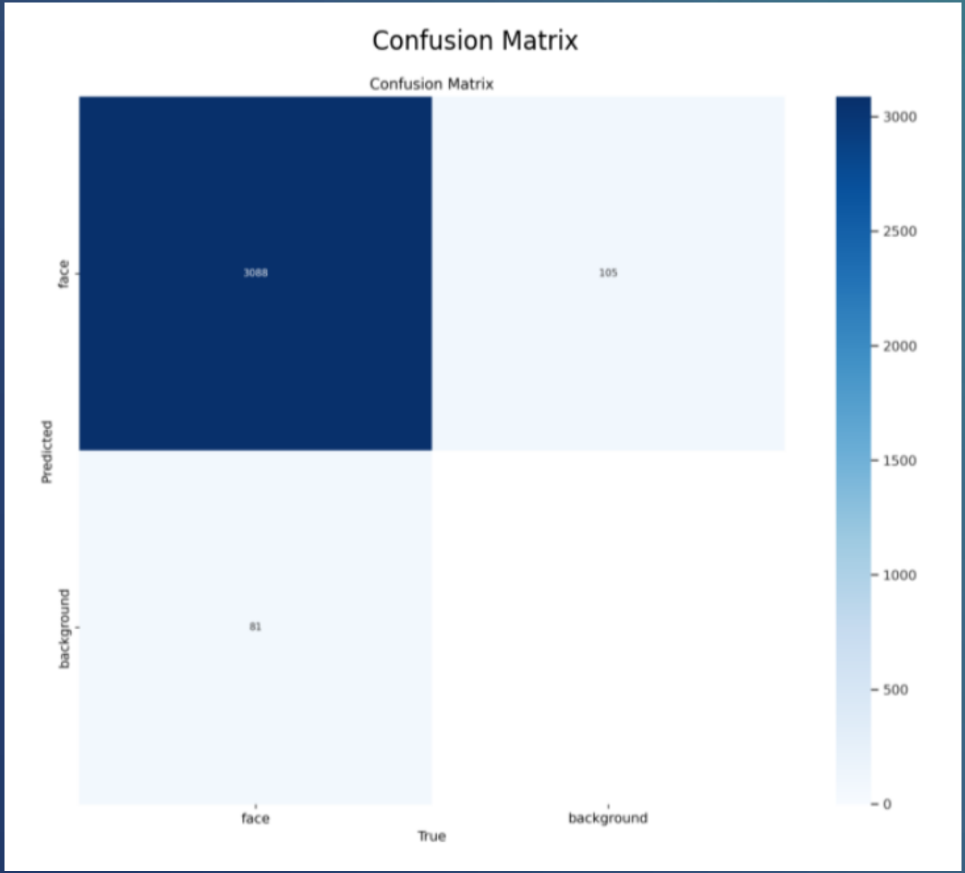
Yann_Martel_0001_jpg.rf.94
Yann_Martel_0002_jpg.rf.49
Yao_Ming_0001_jpg.rf.5196
face

1. YOLOV8 as a Face Detection Model(15 epochs) :

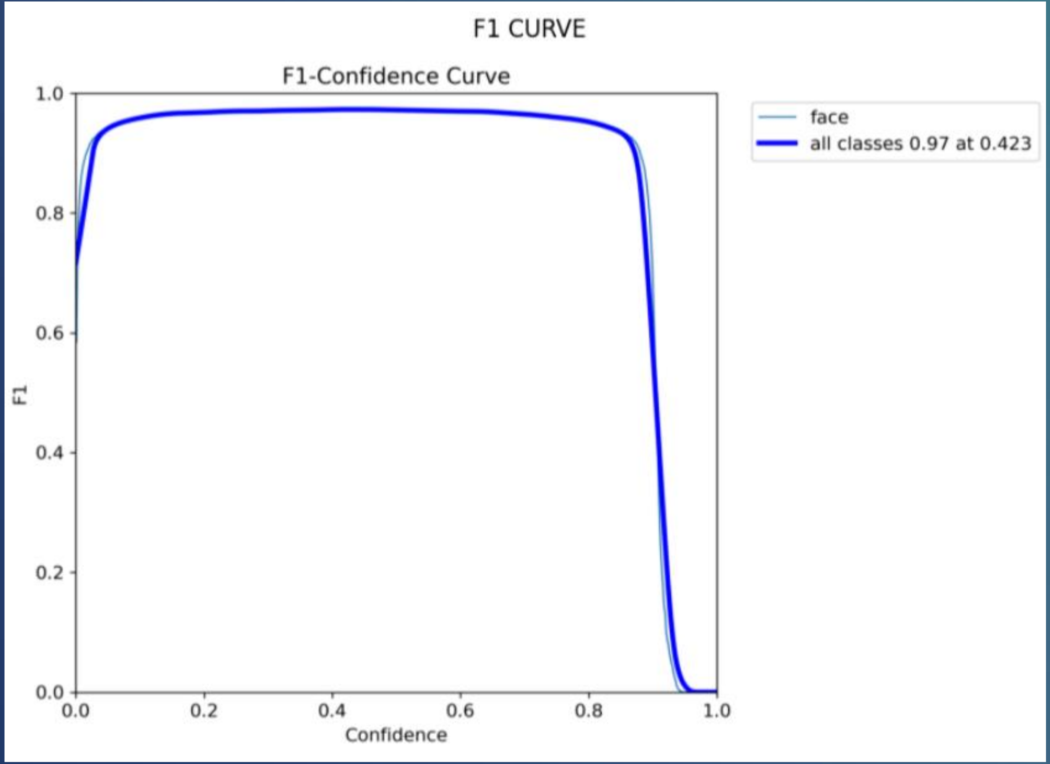
Validation Sample Batch 1 Predictions



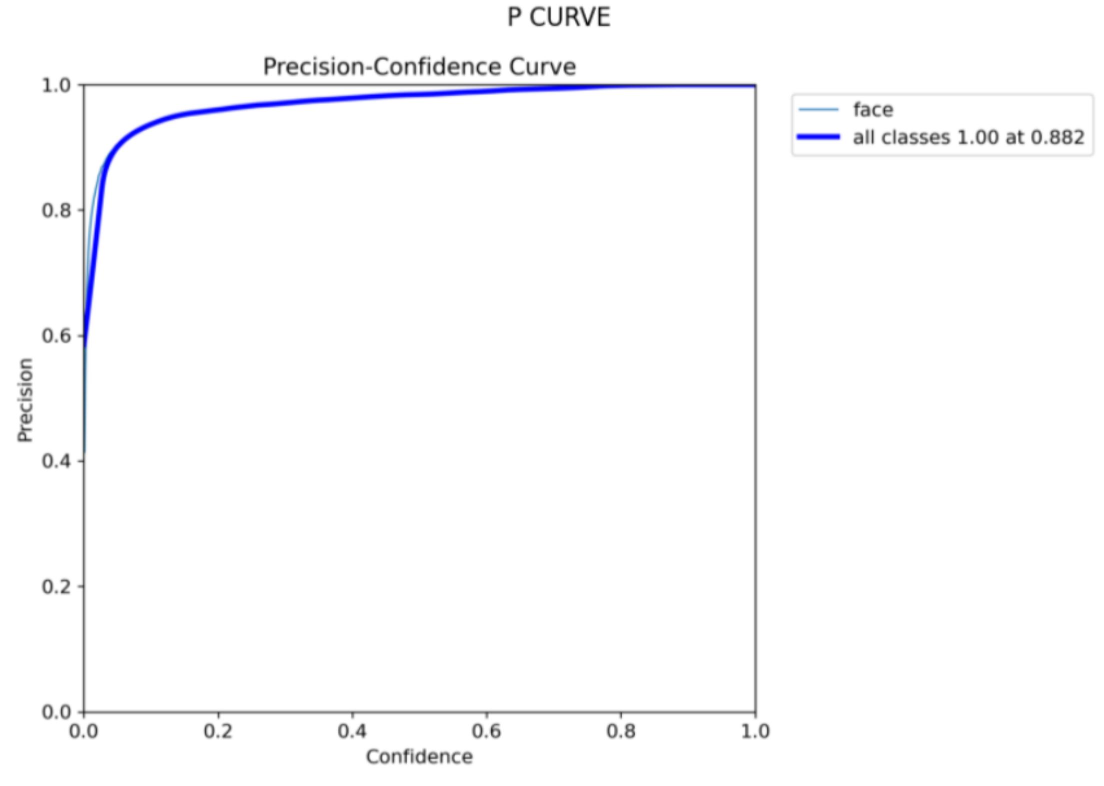
2. YOLOV5 as a Face Detection Model(30 epochs):



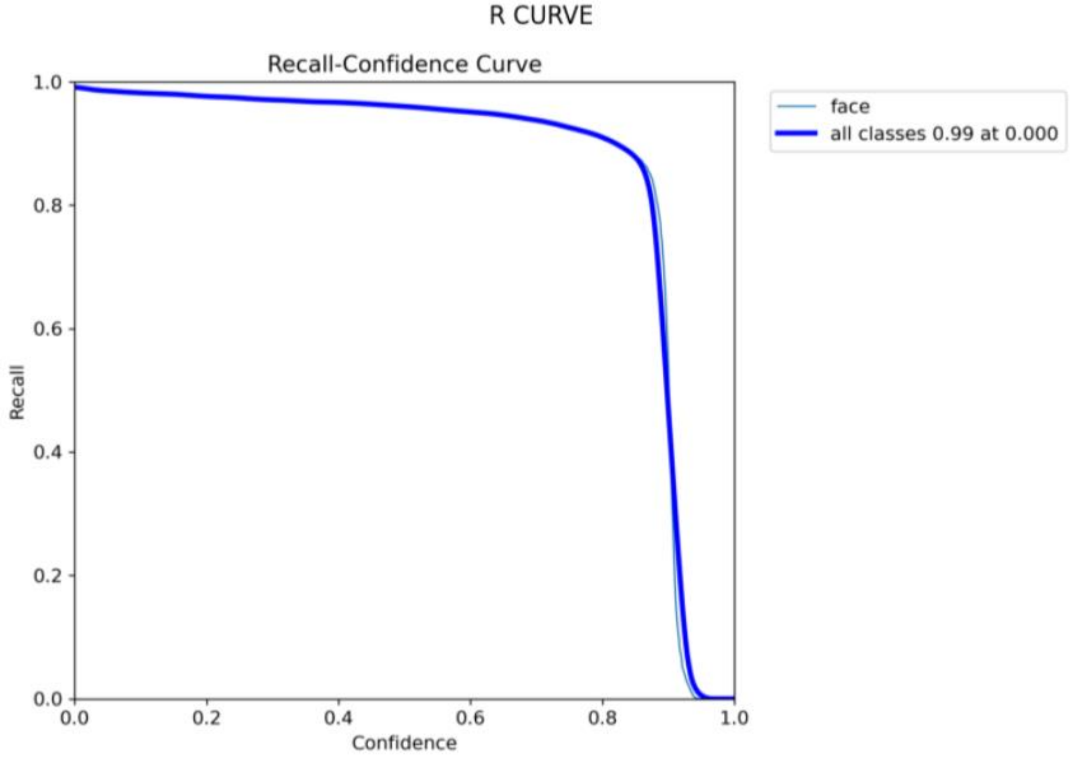
2. YOLOV5 as a Face Detection Model(30 epochs):



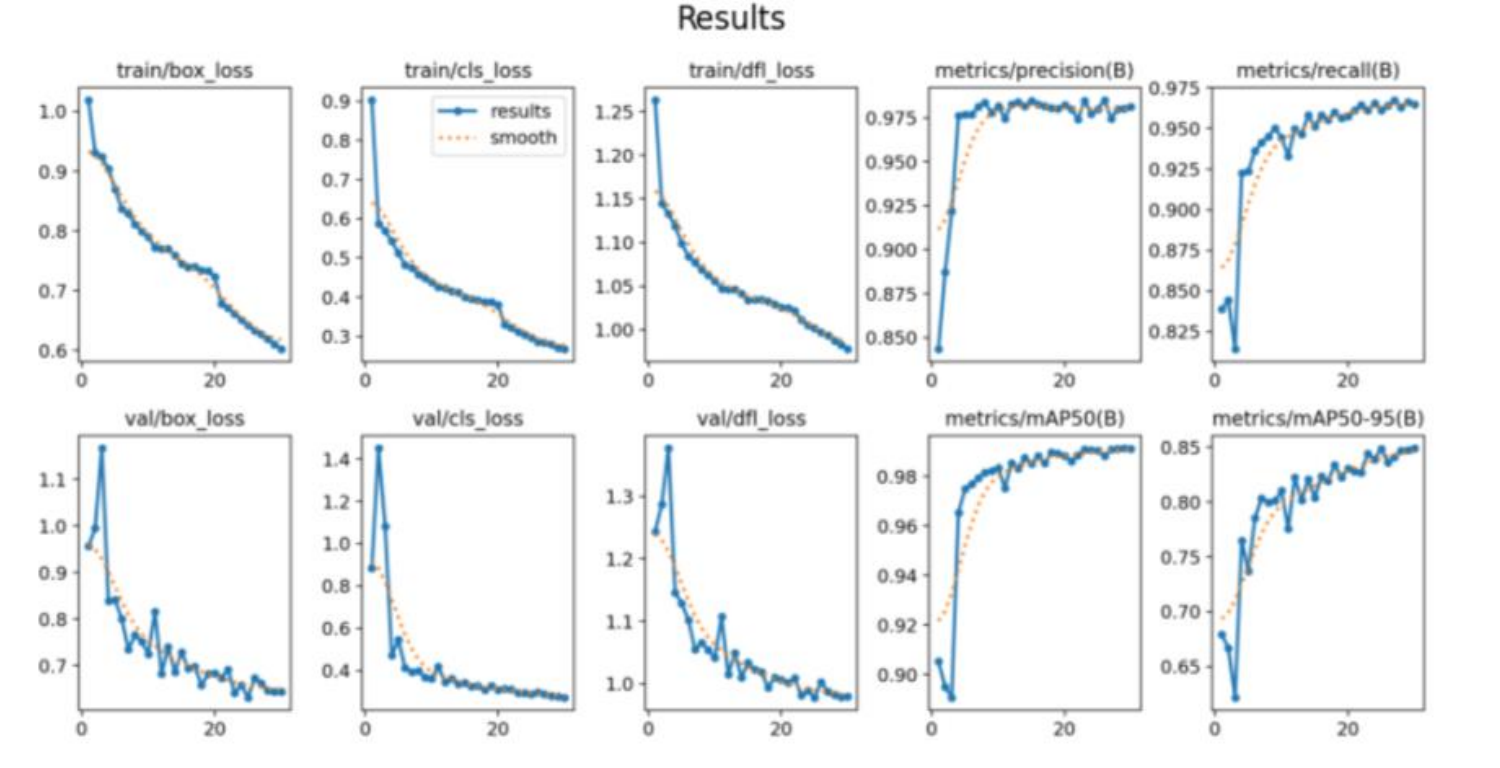
2. YOLOV5 as a Face Detection Model(30 epochs):



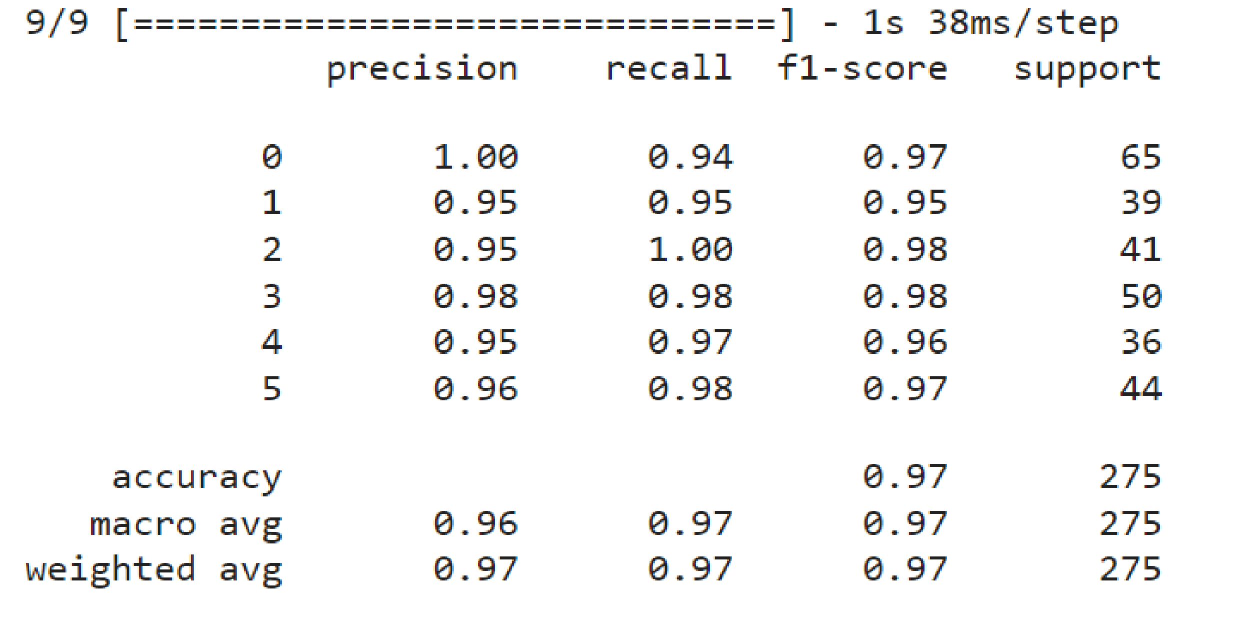
2. YOLOV5 as a Face Detection Model(30 epochs):



2. YOLOV5 as a Face Detection Model(30 epochs):

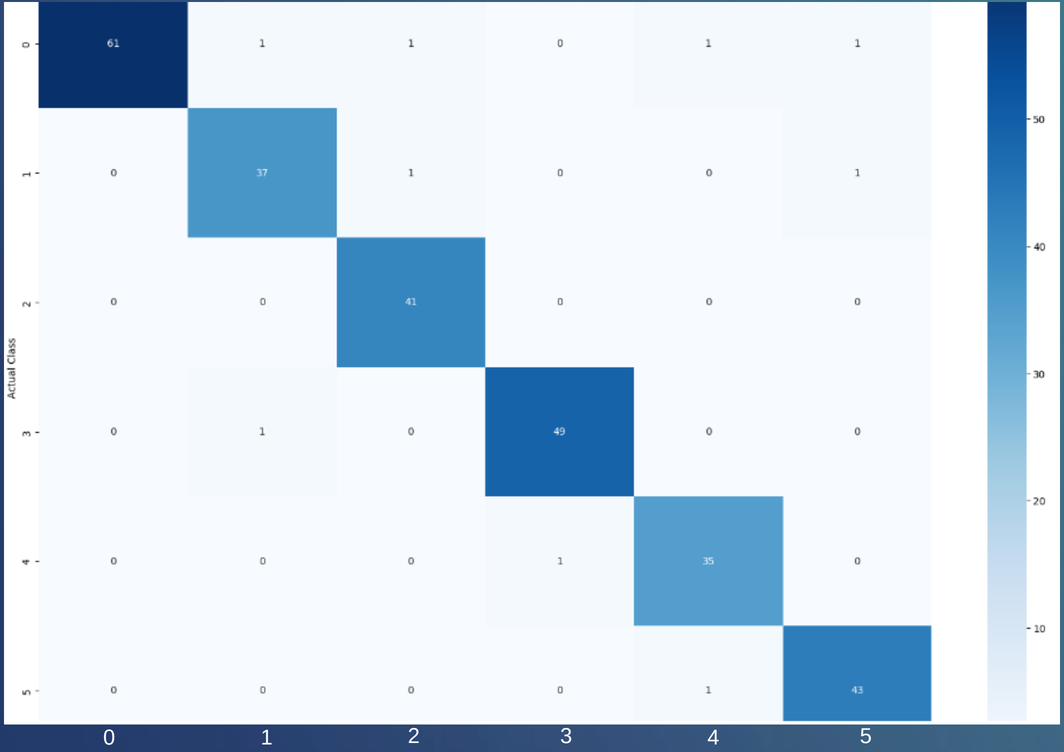


3. MobileNet as a Face Recognition Model(50-35)



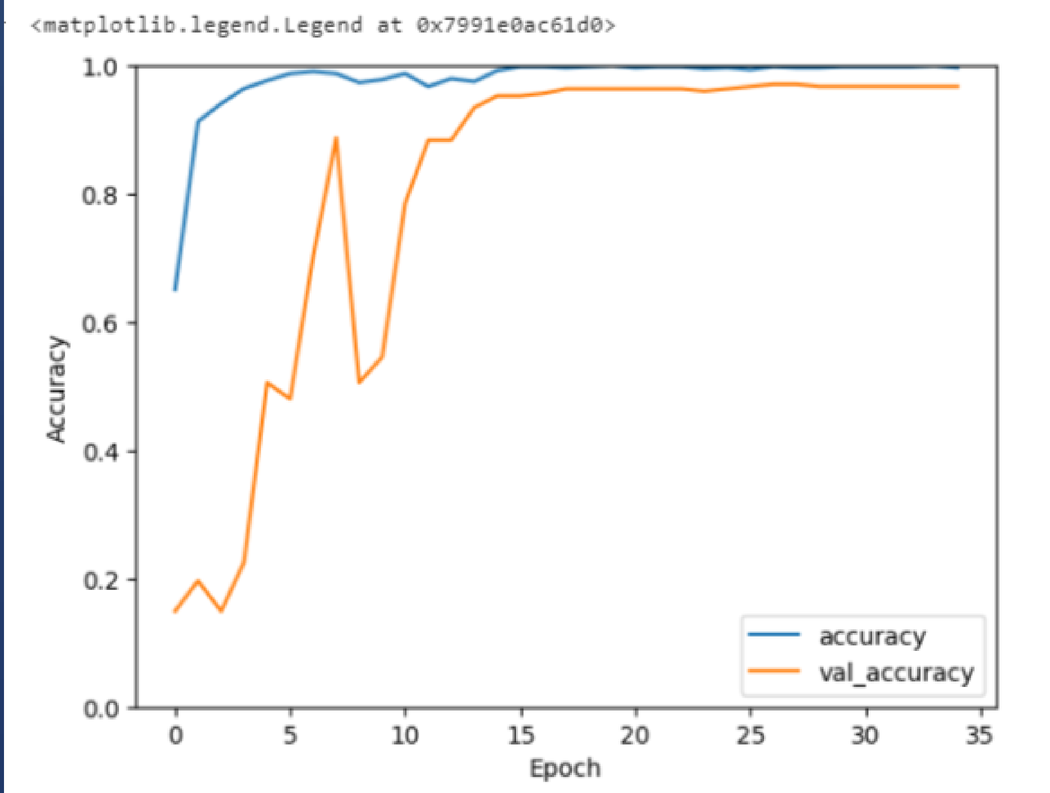
```
9/9 [==============================] - 1s 38ms/step
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        65
           1       0.95      0.95      0.95        39
           2       0.95      1.00      0.98        41
           3       0.98      0.98      0.98        50
           4       0.95      0.97      0.96        36
           5       0.96      0.98      0.97        44

    accuracy                           0.97       275
   macro avg       0.96      0.97      0.97       275
weighted avg       0.97      0.97      0.97       275
```

3. MobileNet as a Face Recognition Model



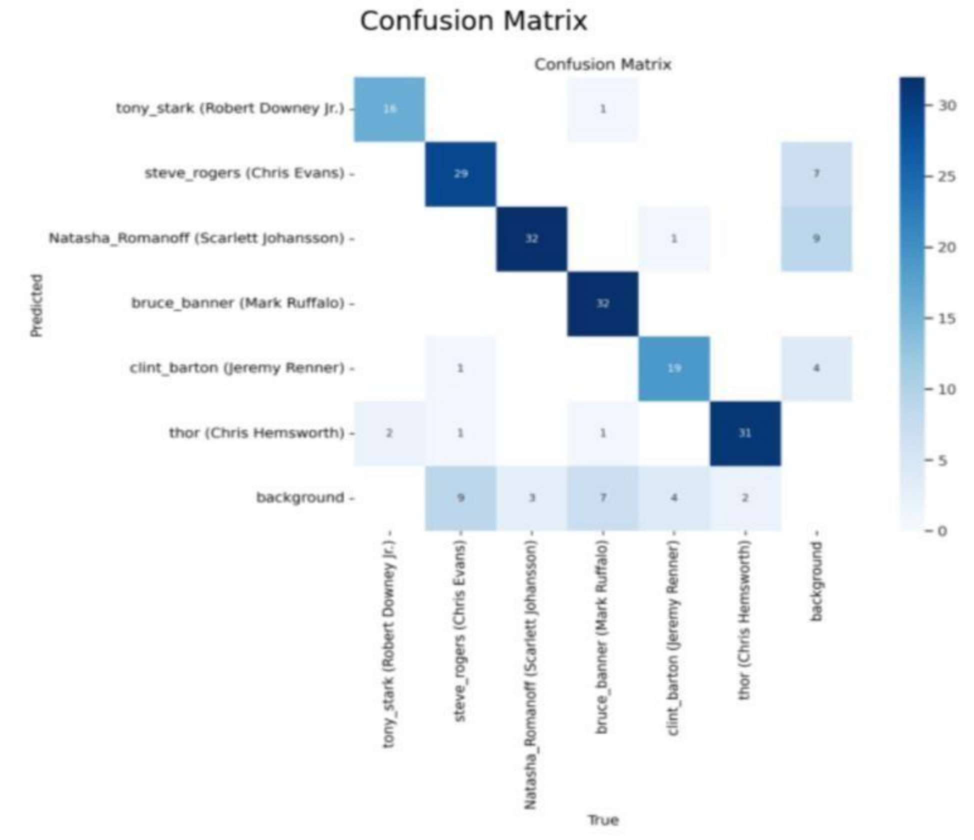
Actual Class	0	1	2	3	4	5
0	61	1	1	0	1	1
1	0	37	1	0	0	1
2	0	0	41	0	0	0
3	0	1	0	49	0	0
4	0	0	0	1	35	0
5	0	0	0	0	1	43

3. MobileNet as a Face Recognition Model



```
<matplotlib.legend.Legend at 0x7991e0ac61d0>
```

4. YoloV5 as a Face Recognition Model(35 epochs)

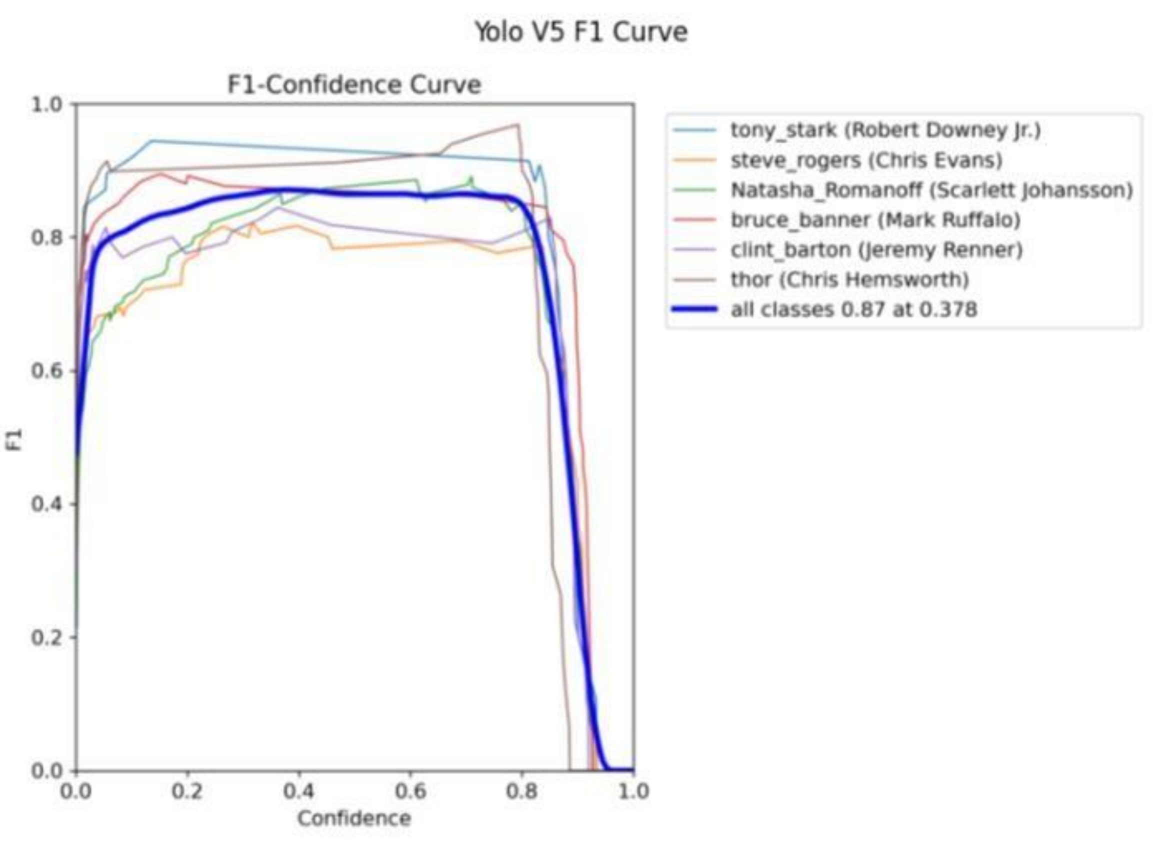


Confusion Matrix

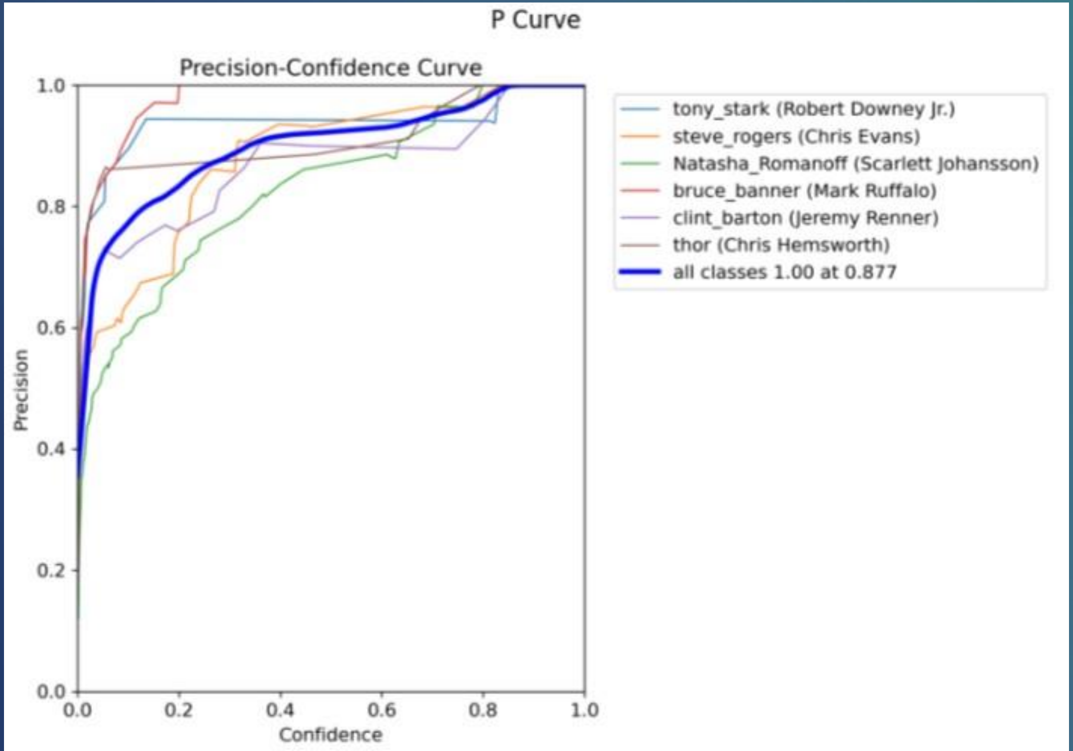
Confusion Matrix

	tony_stark (Robert Downey Jr.)	steve_rogers (Chris Evans)	Natasha_Romanoff (Scarlett Johansson)	bruce_banner (Mark Ruffalo)	clint_barton (Jeremy Renner)	thor (Chris Hemsworth)	background
tony_stark (Robert Downey Jr.)	16			1			
steve_rogers (Chris Evans)		29					7
Natasha_Romanoff (Scarlett Johansson)			32		1		9
bruce_banner (Mark Ruffalo)				32			
clint_barton (Jeremy Renner)		1			19		4
thor (Chris Hemsworth)	2	1		1		31	
background		9	3	7	4	2	

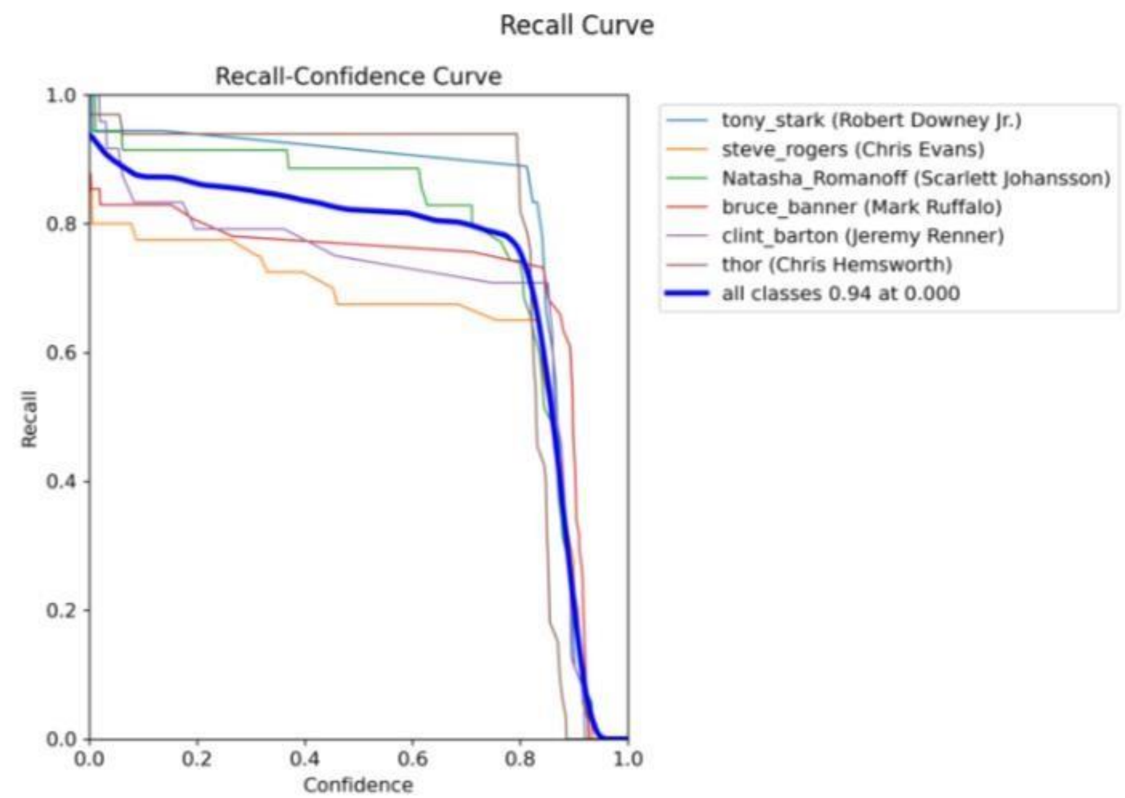
4. YoloV5 as a Face Recognition Model



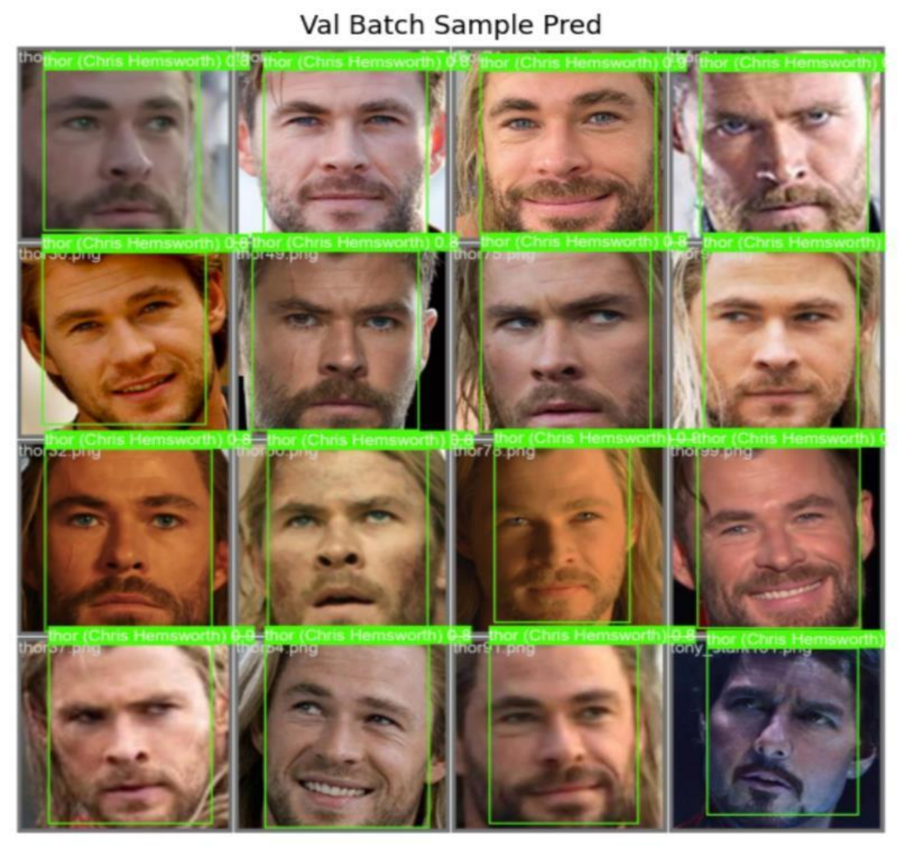
4. YoloV5 as a Face Recognition Model



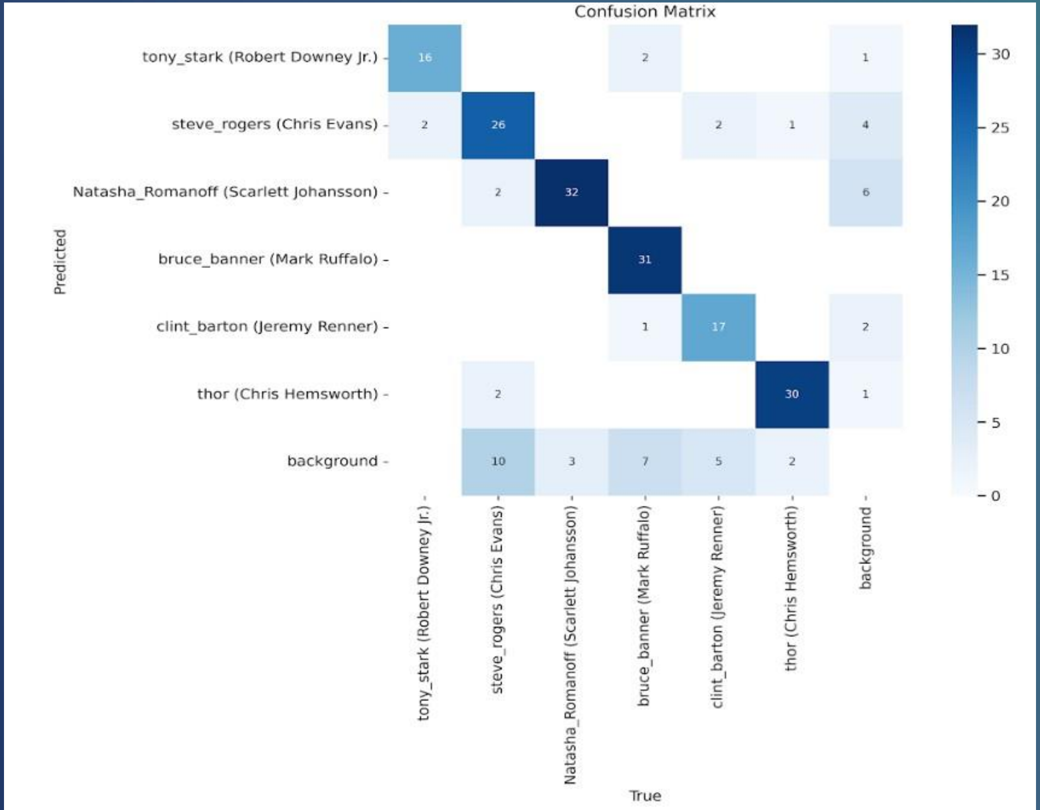
4. YoloV5 as a Face Recognition Model



4. YoloV5 as a Face Recognition Model


Val Batch Sample Pred
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.9
thor (Chris Hemsworth)
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth)
thor49.png
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth)
thor32.png
thor78.png
thor99.png
thor (Chris Hemsworth) 0.9
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth) 0.8
thor (Chris Hemsworth)
thor91.png

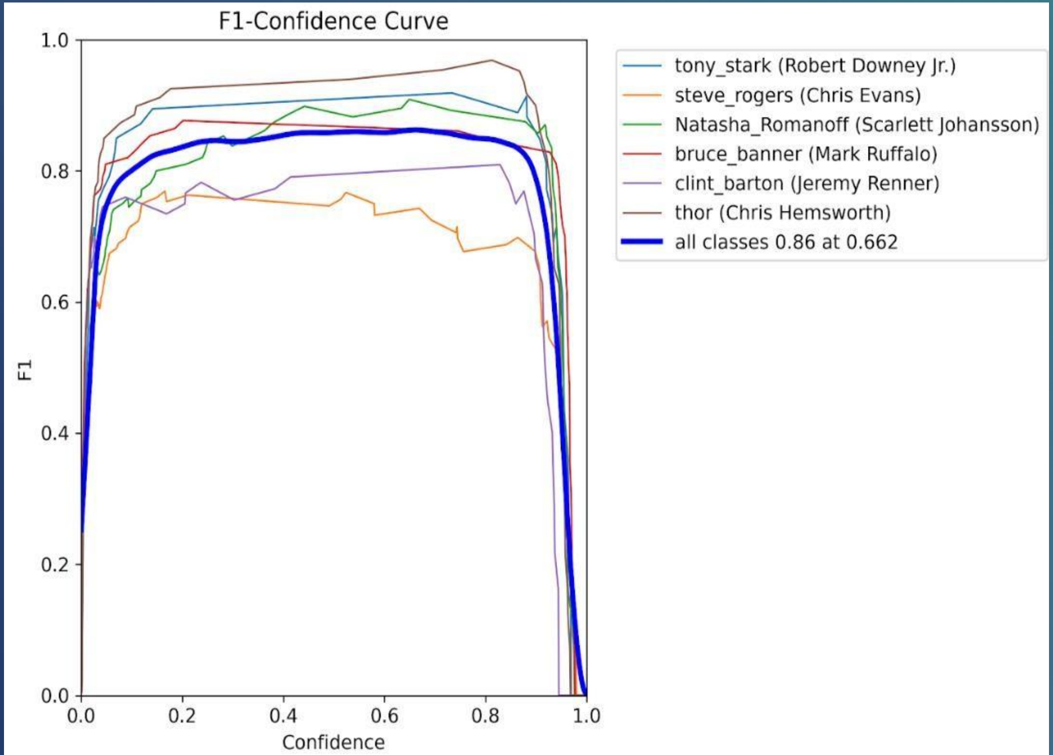
5. YoloV8 as a Face Recognition Model (25 epochs)



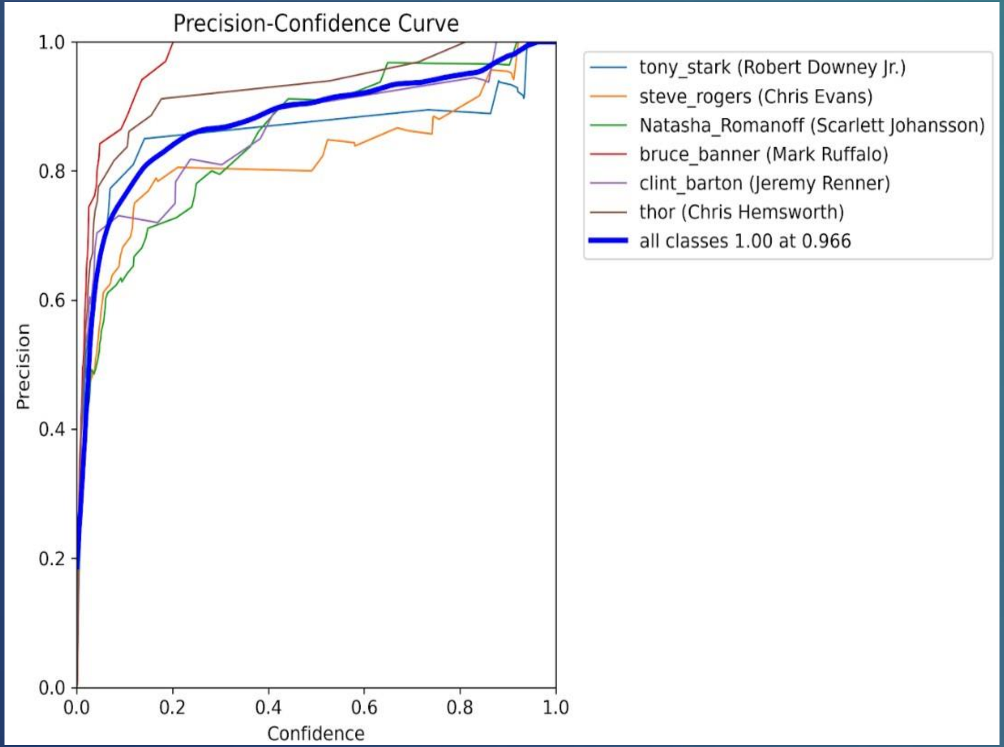
Confusion Matrix

	tony_stark (Robert Downey Jr.)	steve_rogers (Chris Evans)	Natasha_Romanoff (Scarlett Johansson)	bruce_banner (Mark Ruffalo)	clint_barton (Jeremy Renner)	thor (Chris Hemsworth)	background
tony_stark (Robert Downey Jr.)	16			2			1
steve_rogers (Chris Evans)	2	26			2	1	4
Natasha_Romanoff (Scarlett Johansson)		2	32				6
bruce_banner (Mark Ruffalo)				31			
clint_barton (Jeremy Renner)				1	17		2
thor (Chris Hemsworth)		2				30	1
background		10	3	7	5	2	

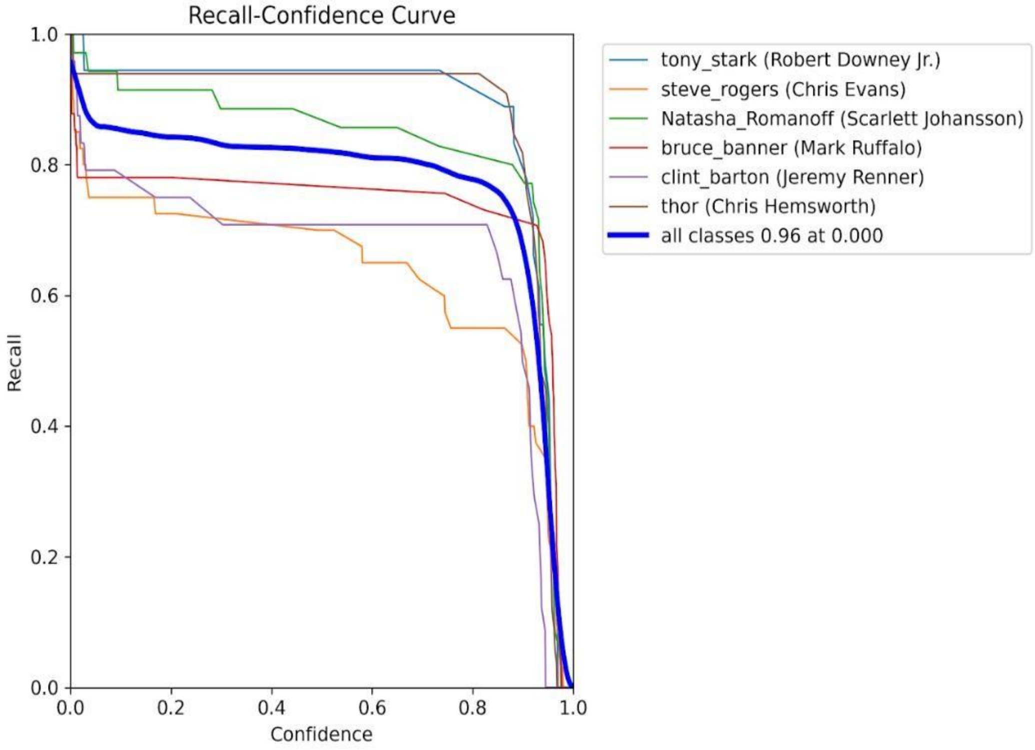
5. YoloV8 as a Face Recognition Model



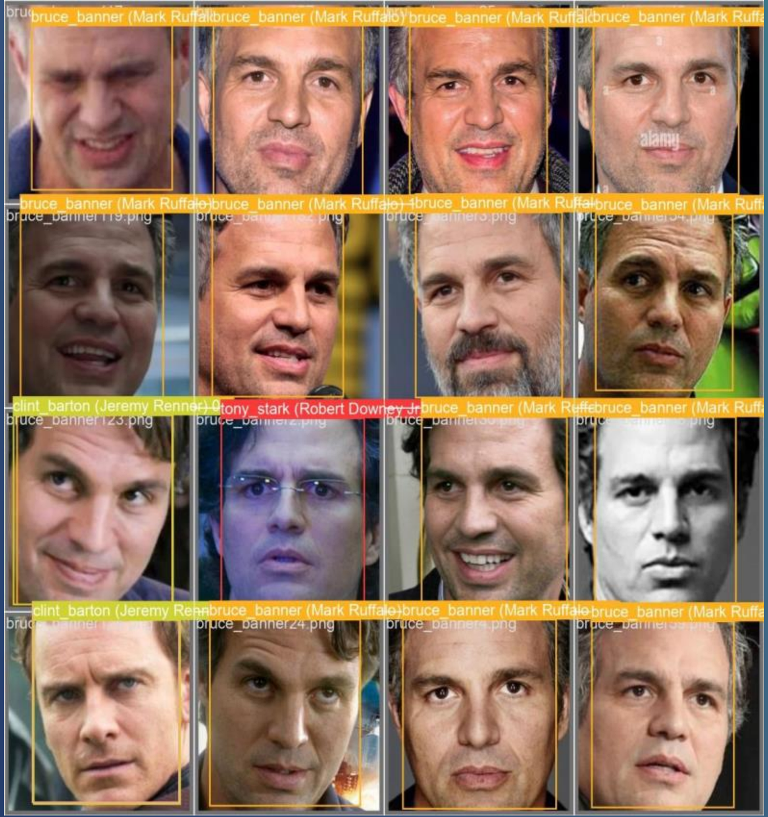
5. YoloV8 as a Face Recognition Model



5. YoloV8 as a Face Recognition Model



5. YoloV8 as a Face Recognition Model



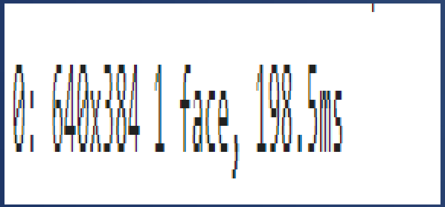
Validation Predictions Samples

12

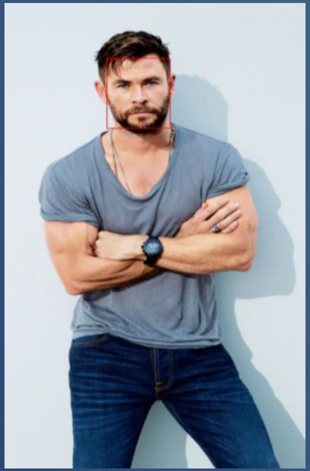
Results

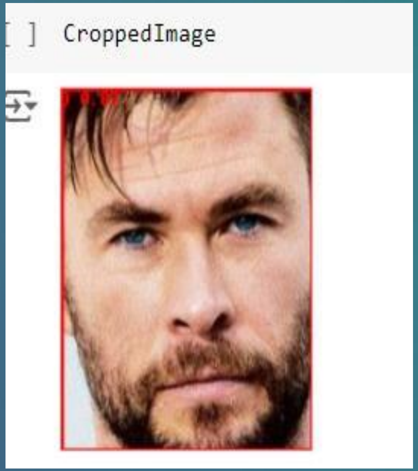
YoloV8 Face Detection and Recognition Models

Results of an imagine with a :
single face



```
0: 640x384 1 face, 198.5ms
```

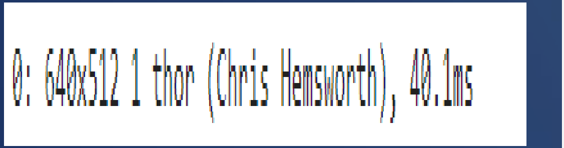




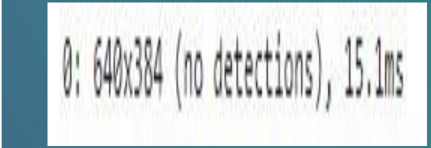
```
[ ] CroppedImage
```

YoloV8 Face Detection and Recognition Models

Results of an imagine with a :
single face



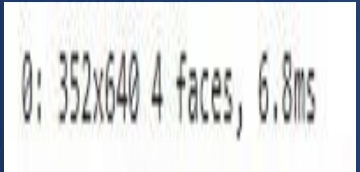
```
0: 640x512 1 thor (Chris Hemsworth), 40.1ms
```



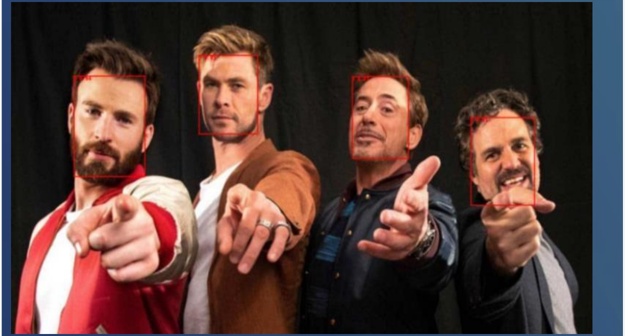
```
0: 640x384 (no detections), 15.1ms
```

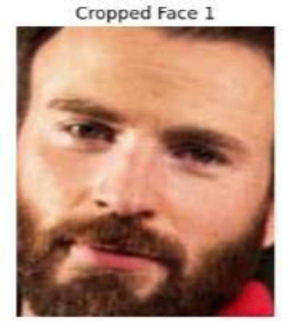
YoloV8 Face Detection and Recognition Models

Results of an imagine with :
Multiple faces



```
0: 352x640 4 faces, 6.8ms
```

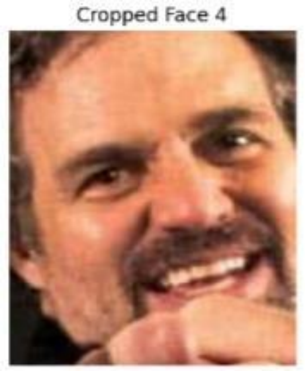




Cropped Face 1

Cropped Face 2

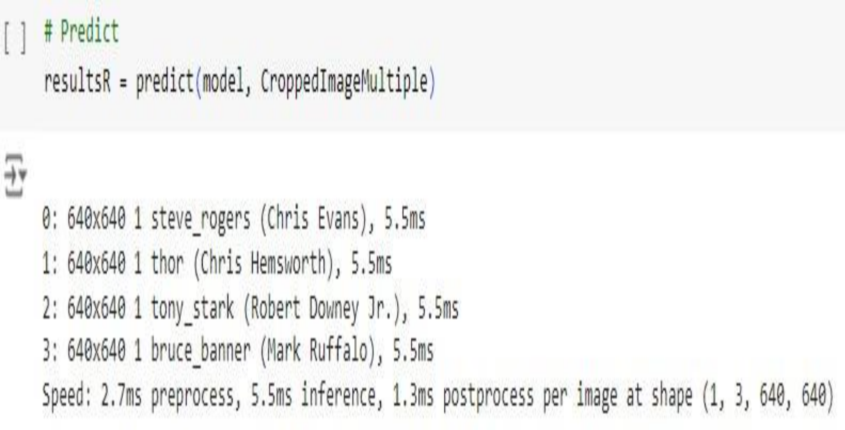
Cropped Face 3



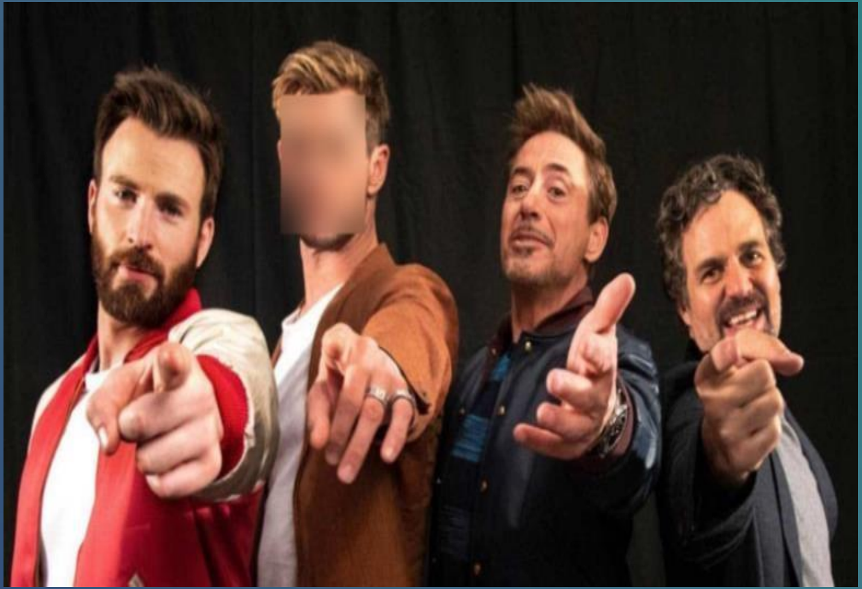
Cropped Face 4

YoloV8 Face Detection and Recognition Models

Results of an imagine with :
Multiple faces

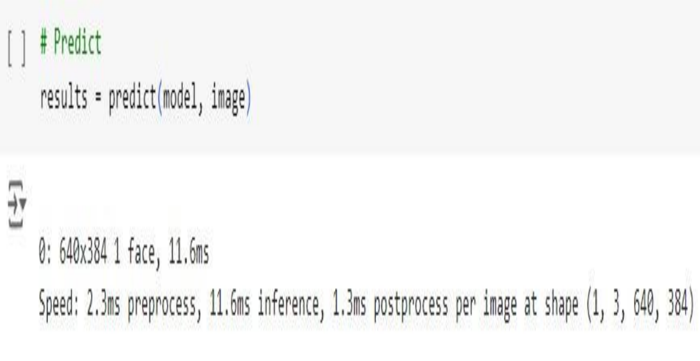


```
[ ] # Predict
    resultsR = predict(model, CroppedImageMultiple)

0: 640x640 1 steve_rogers (Chris Evans), 5.5ms
1: 640x640 1 thor (Chris Hemsworth), 5.5ms
2: 640x640 1 tony_stark (Robert Downey Jr.), 5.5ms
3: 640x640 1 bruce_banner (Mark Ruffalo), 5.5ms
Speed: 2.7ms preprocess, 5.5ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
```



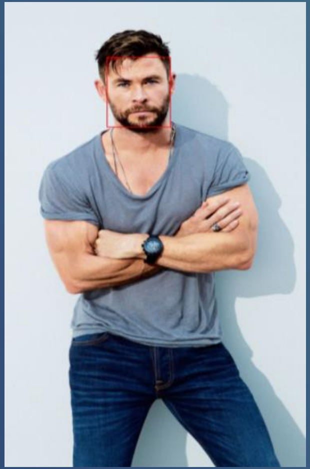
YoloV5 Face Detection and Recognition Models

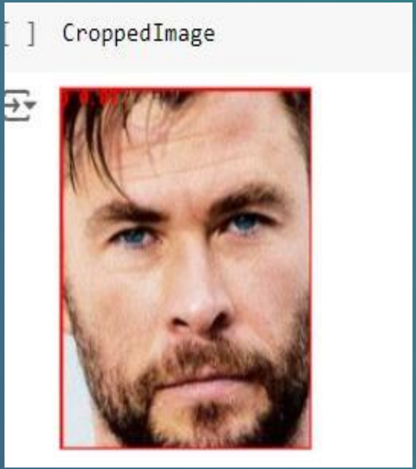
Results of an imagine with a :
single face



```
# Predict
results = predict(model, image)
```

```
0: 640x384 1 face, 11.6ms
Speed: 2.3ms preprocess, 11.6ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 384)
```

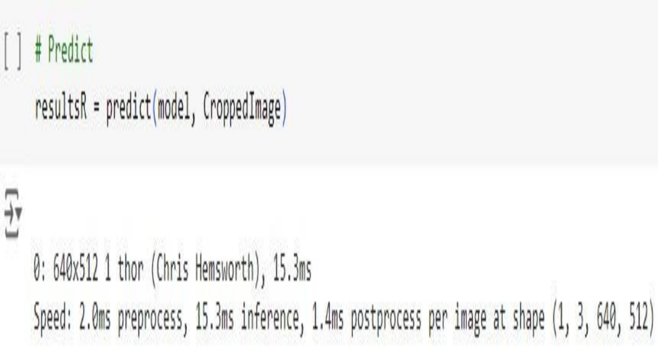




```
CroppedImage
```

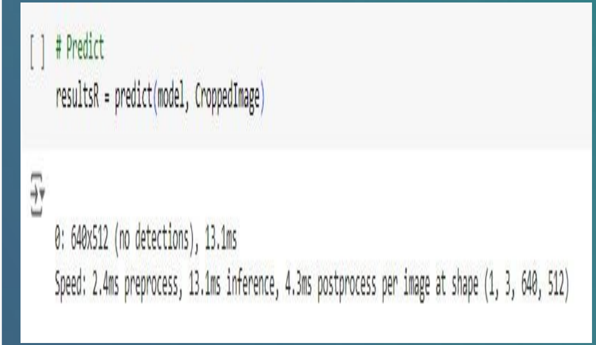
YoloV5 Face Detection and Recognition Models

Results of an imagine with a :
single face



```
# Predict
resultsR = predict(model, CroppedImage)
```

```
0: 640x512 1 thor (Chris Hemsworth), 15.3ms
Speed: 2.0ms preprocess, 15.3ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 512)
```

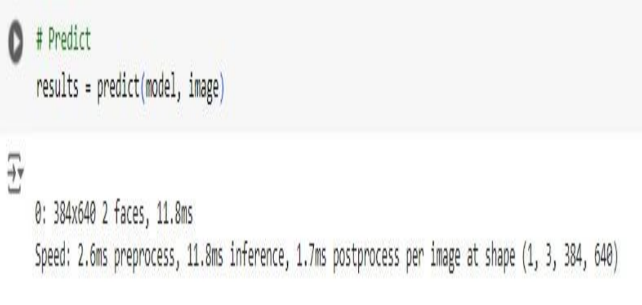


```
# Predict
resultsR = predict(model, CroppedImage)
```

```
0: 640x512 (no detections), 13.1ms
Speed: 2.4ms preprocess, 13.1ms inference, 4.3ms postprocess per image at shape (1, 3, 640, 512)
```

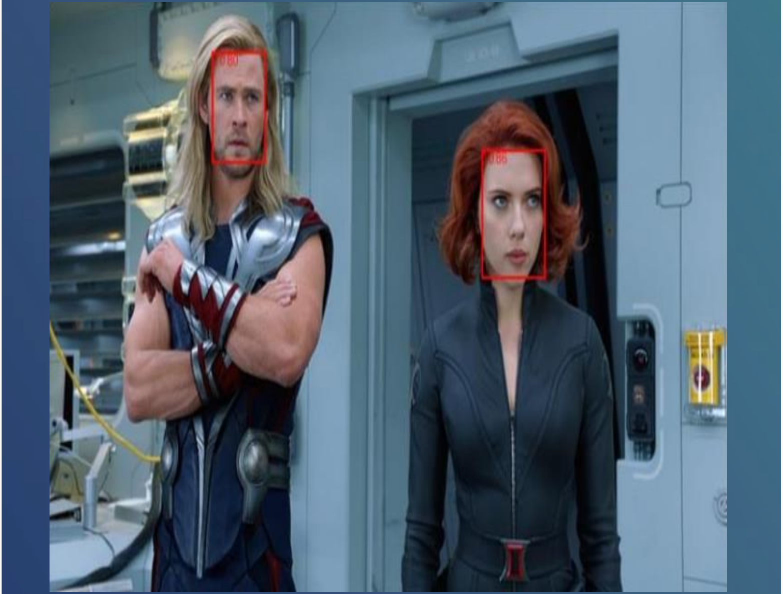
YoloV5 Face Detection and Recognition Models

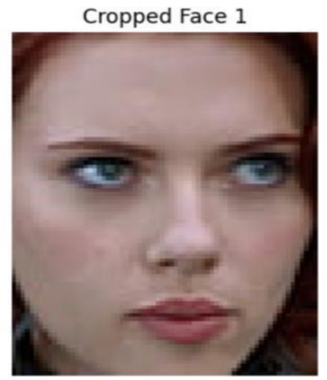
Results of an imagine with :
Multiple faces



```
# Predict
results = predict(model, image)
```

```
0: 384x640 2 faces, 11.8ms
Speed: 2.6ms preprocess, 11.8ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)
```



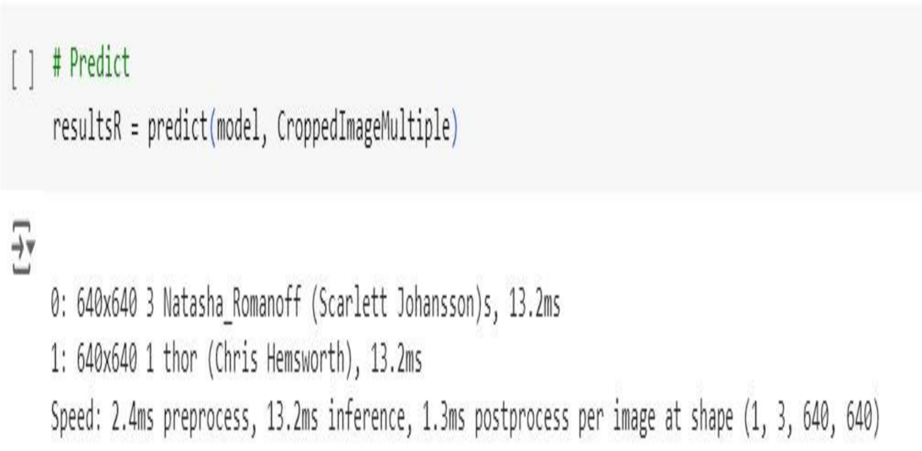


Cropped Face 1

Cropped Face 2

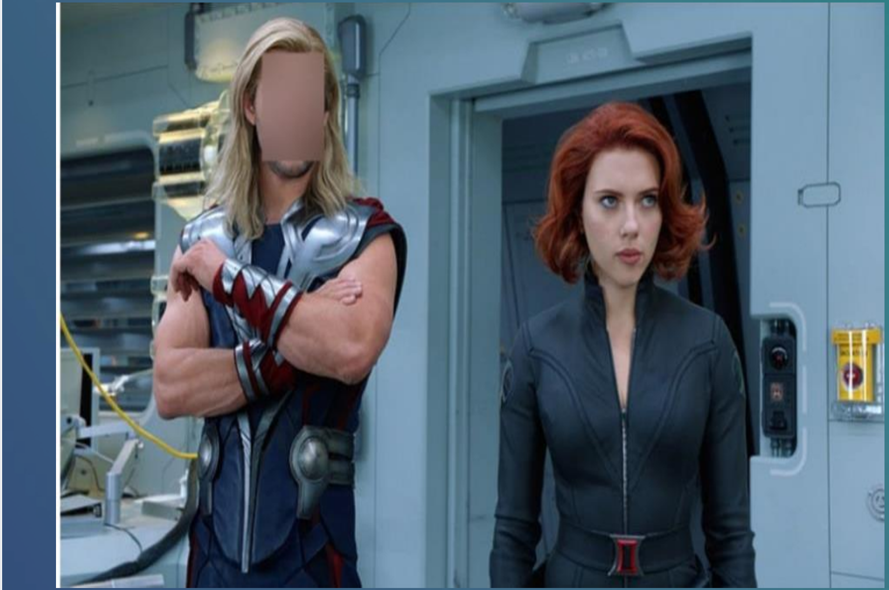
YoloV5 Face Detection and Recognition Models

Results of an imagine with :
Multiple faces



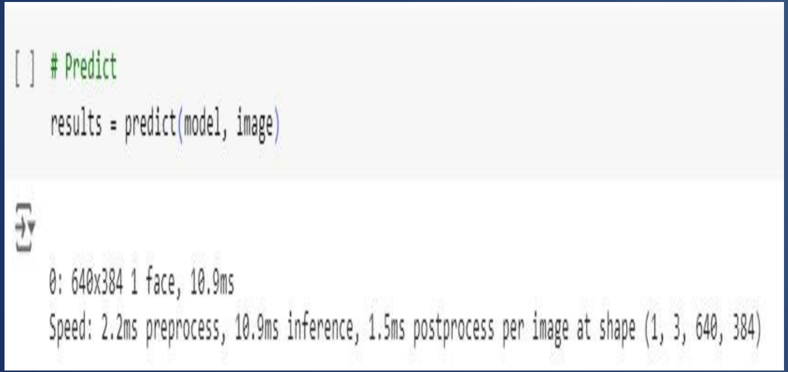
```
# Predict
resultsR = predict(model, CroppedImageMultiple)
```

```
0: 640x640 3 Natasha_Romanoff (Scarlett Johansson)s, 13.2ms
1: 640x640 1 thor (Chris Hemsworth), 13.2ms
Speed: 2.4ms preprocess, 13.2ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
```



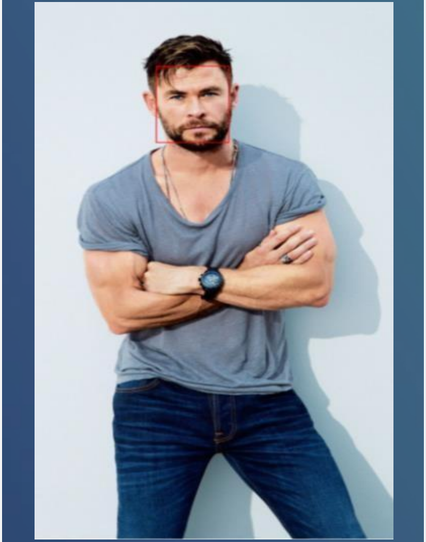
The YOLOV8 For Face Detection and MobileNet for Face Recognition

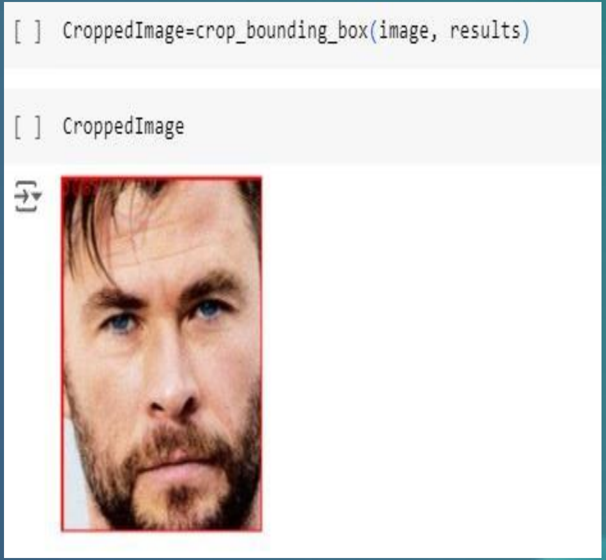
Results of an imagine with a :
single face



```
# Predict
results = predict(model, image)
```

```
0: 640x384 1 face, 10.9ms
Speed: 2.2ms preprocess, 10.9ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 384)
```



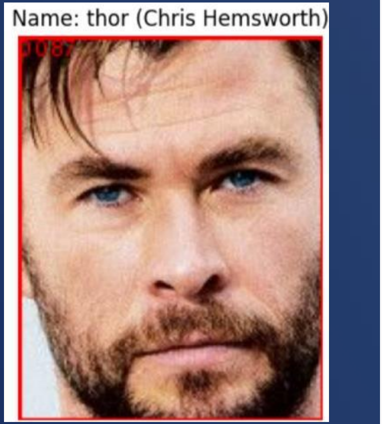


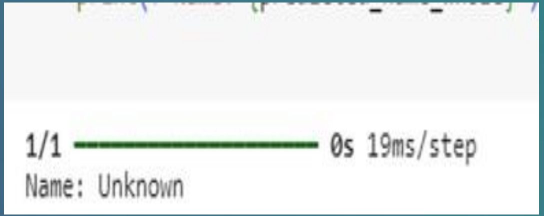
```
CroppedImage=crop_bounding_box(image, results)
```

```
CroppedImage
```

The YOLOV8 For Face Detection and MobileNet for Face Recognition

Results of an imagine with a :
single face

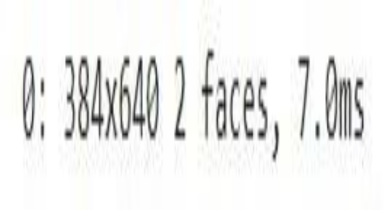




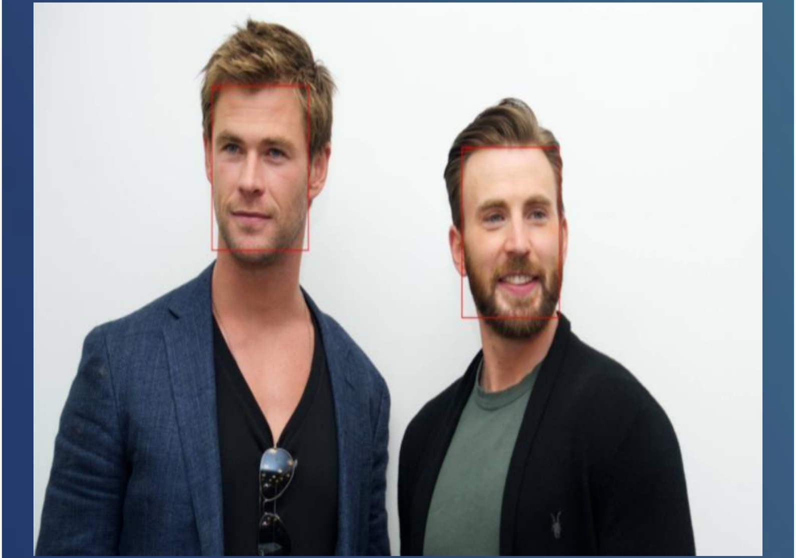
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Name: Unknown
```

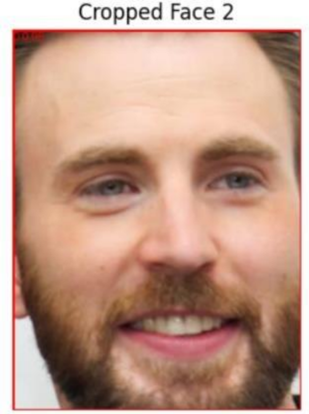
The YOLOV8 For Face Detection and MobileNet for Face Recognition

Results of an imagine with :
Multiple faces



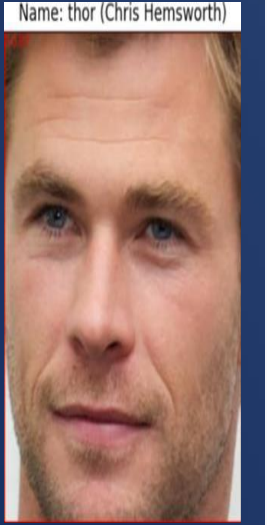
```
0: 384x640 2 faces, 7.0ms
```

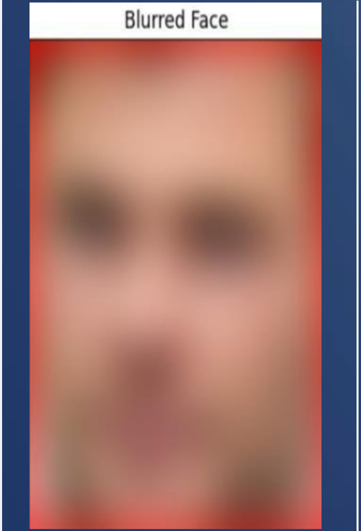


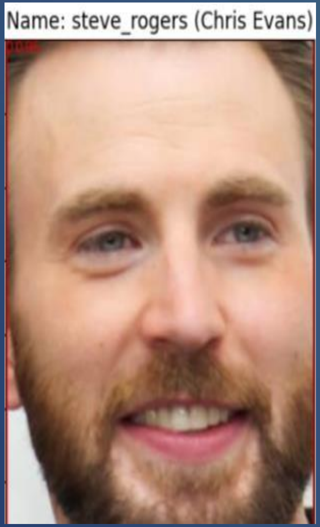


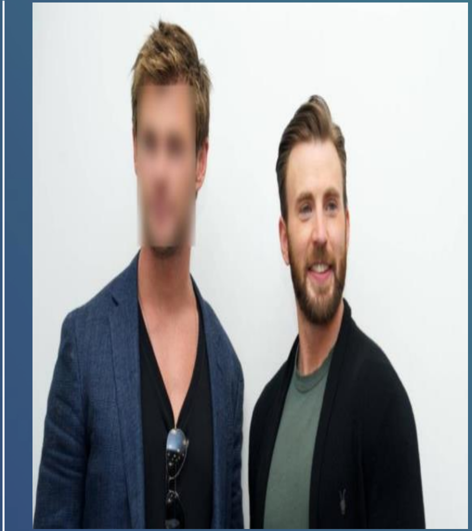
The YOLOV8 For Face Detection and MobileNet for Face Recognition

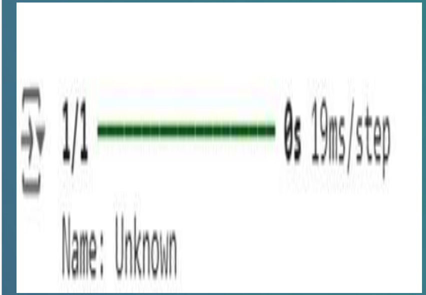
Results of an imagine with :
Multiple faces







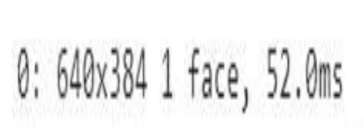




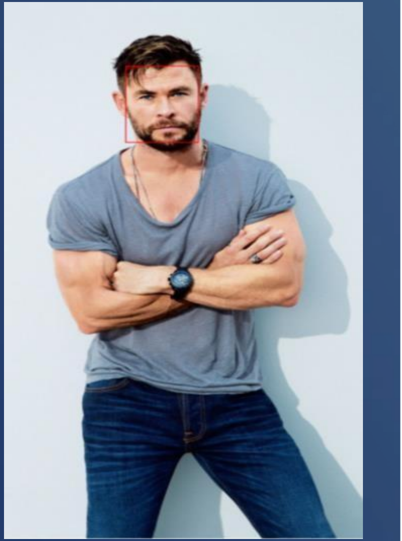
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Name: Unknown
```

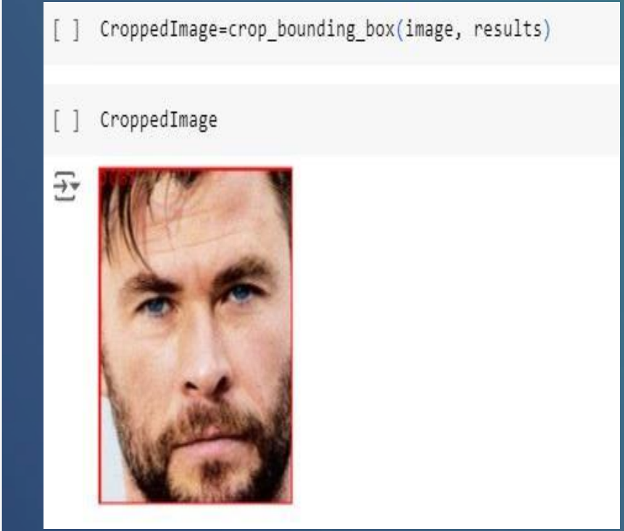
The YOLOV5 For Face Detection and MobileNet for Face Recognition

Results of an imagine with a :
Single face

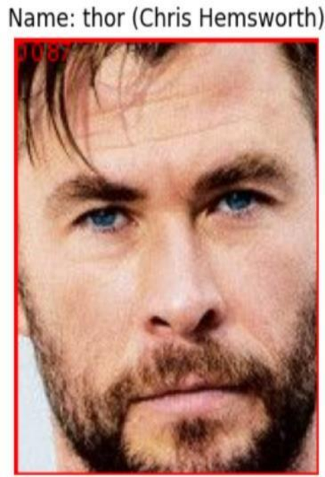


```
0: 640x384 1 face, 52.0ms
```





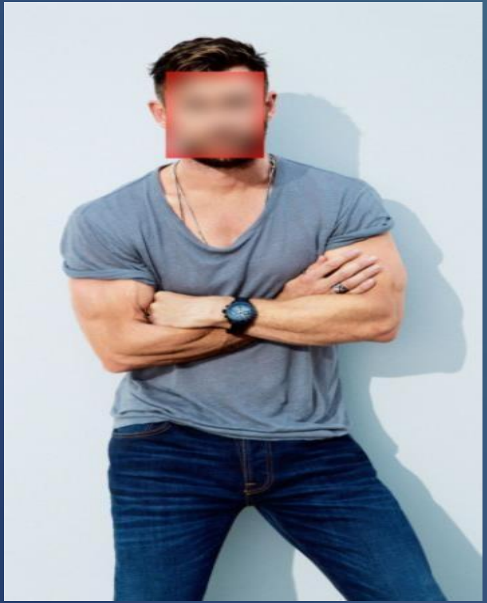
```
[ ] CroppedImage=crop_bounding_box(image, results)

[ ] CroppedImage
```

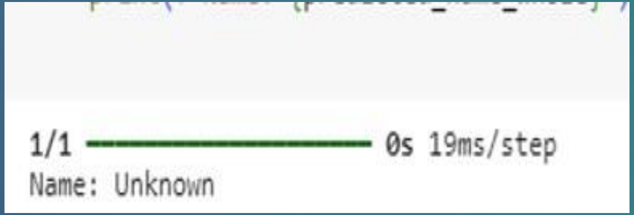


Name: thor (Chris Hemsworth)

The YOLOV5 For Face Detection and MobileNet for Face Recognition

Results of an imagine with a :
Single face

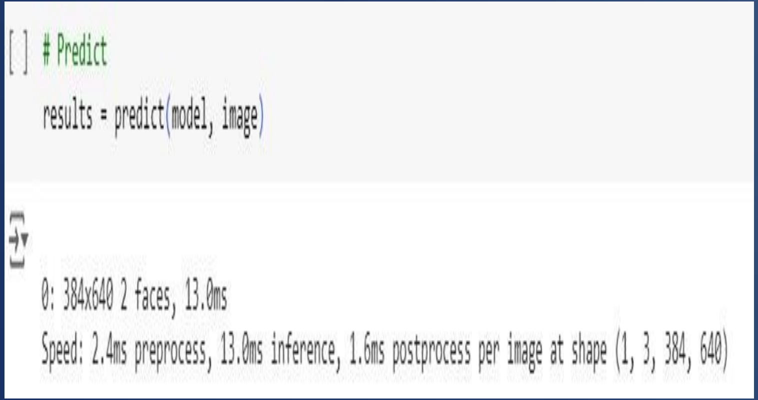




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Name: Unknown
```

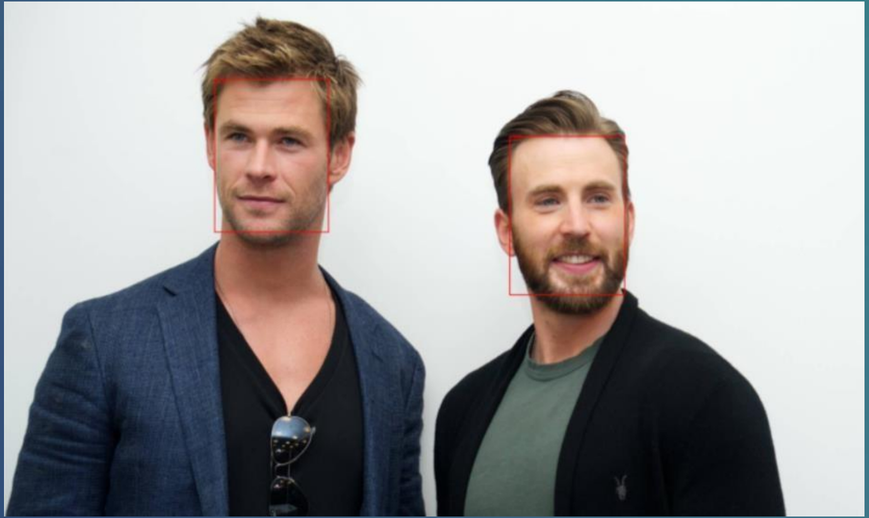
The YOLOV5 For Face Detection and MobileNet for Face Recognition

Results of an imagine with :
Multiple faces



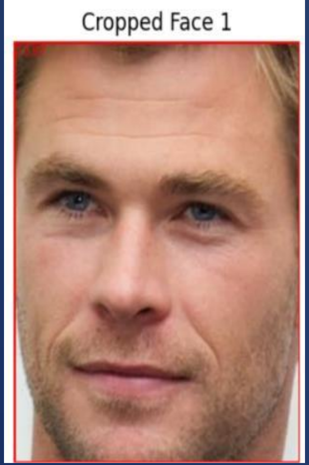
```
[ ] # Predict
    results = predict(model, image)
```

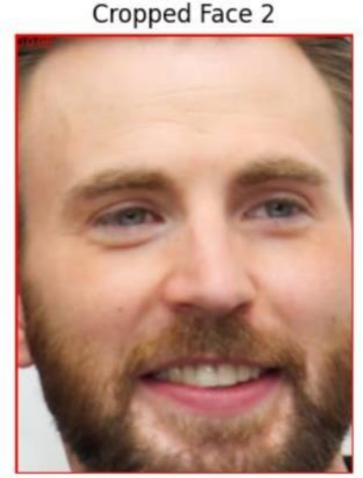
```
0: 384x640 2 faces, 13.0ms
Speed: 2.4ms preprocess, 13.0ms inference, 1.6ms postprocess per image at shape (1, 3, 384, 640)
```

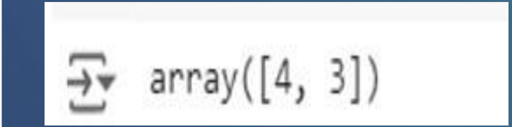


The YOLOV5 For Face Detection and MobileNet for Face Recognition

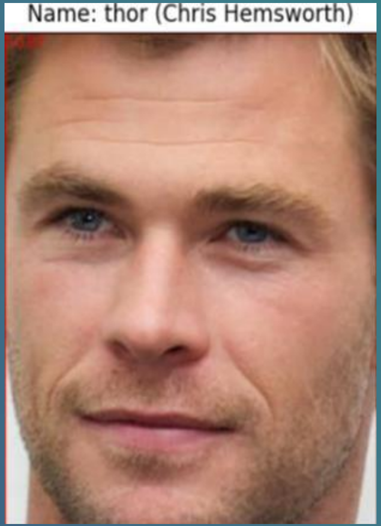
Results of an imagine with :
Multiple faces





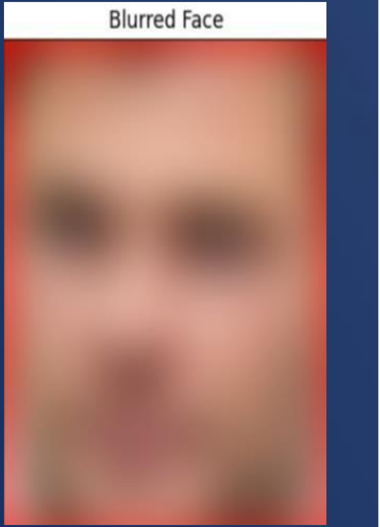


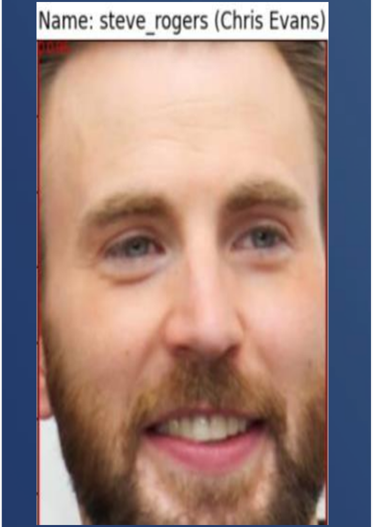
```
array([4, 3])
```

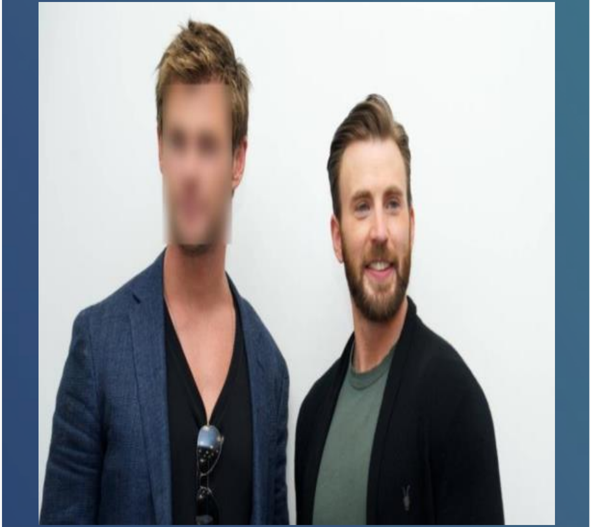


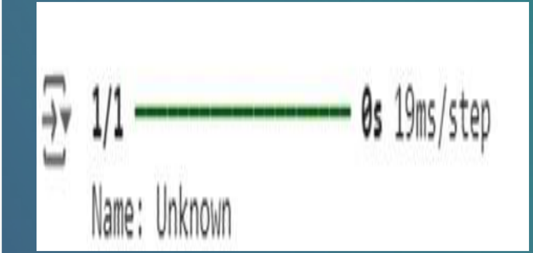
The YOLOV5 For Face Detection and MobileNet for Face Recognition

Results of an imagine with :
Multiple faces



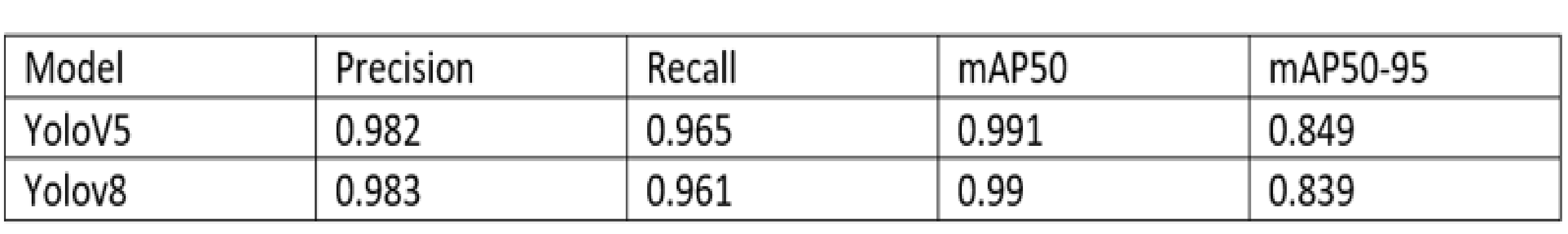






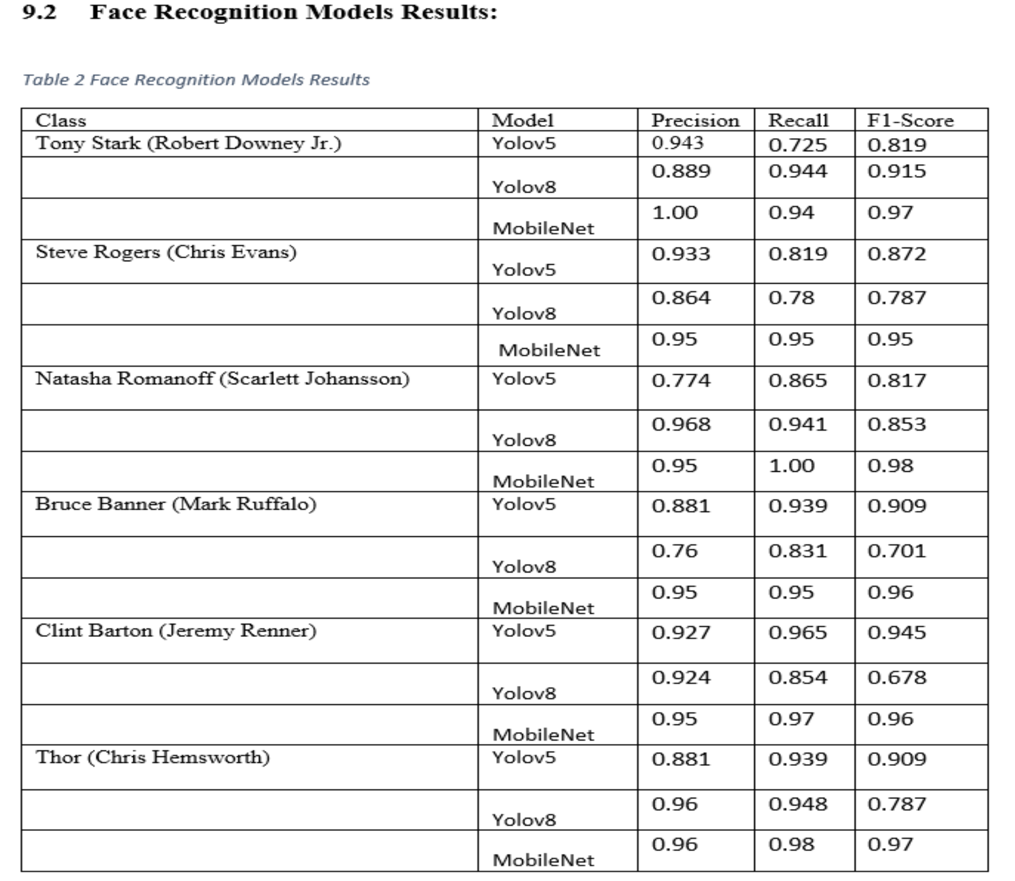
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Name: Unknown
```

Face Detection Models Results



Model	Precision	Recall	mAP50	mAP50-95
YoloV5	0.982	0.965	0.991	0.849
Yolov8	0.983	0.961	0.99	0.839

Face Recognition Models Results

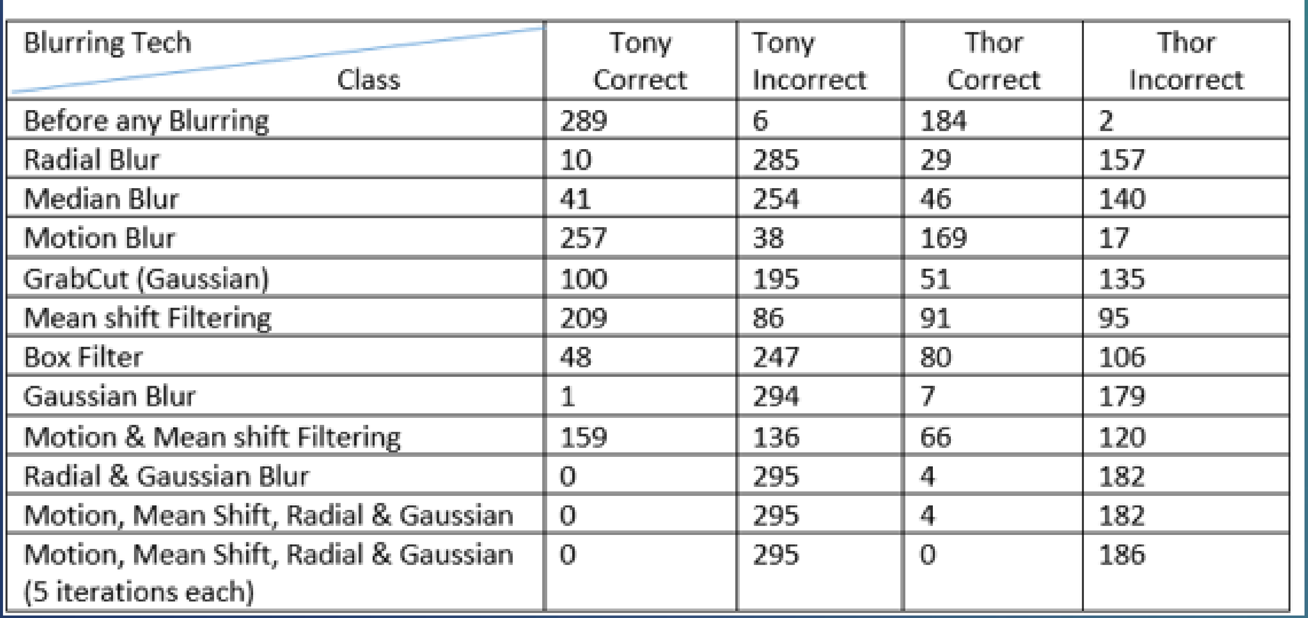


9.2 Face Recognition Models Results:

Table 2 Face Recognition Models Results

Class	Model	Precision	Recall	F1-Score
Tony Stark (Robert Downey Jr.)	Yolov5	0.943	0.725	0.819
	Yolov8	0.889	0.944	0.915
	MobileNet	1.00	0.94	0.97
Steve Rogers (Chris Evans)	Yolov5	0.933	0.819	0.872
	Yolov8	0.864	0.78	0.787
	MobileNet	0.95	0.95	0.95
Natasha Romanoff (Scarlett Johansson)	Yolov5	0.774	0.865	0.817
	Yolov8	0.968	0.941	0.853
	MobileNet	0.95	1.00	0.98
Bruce Banner (Mark Ruffalo)	Yolov5	0.881	0.939	0.909
	Yolov8	0.76	0.831	0.701
	MobileNet	0.95	0.95	0.96
Clint Barton (Jeremy Renner)	Yolov5	0.927	0.965	0.945
	Yolov8	0.924	0.854	0.678
	MobileNet	0.95	0.97	0.96
Thor (Chris Hemsworth)	Yolov5	0.881	0.939	0.909
	Yolov8	0.96	0.948	0.787
	MobileNet	0.96	0.98	0.97

Recognition Results Before and after Blurring Techniques



Blurring Tech Class	Tony Correct	Tony Incorrect	Thor Correct	Thor Incorrect
Before any Blurring	289	6	184	2
Radial Blur	10	285	29	157
Median Blur	41	254	46	140
Motion Blur	257	38	169	17
GrabCut (Gaussian)	100	195	51	135
Mean shift Filtering	209	86	91	95
Box Filter	48	247	80	106
Gaussian Blur	1	294	7	179
Motion & Mean shift Filtering	159	136	66	120
Radial & Gaussian Blur	0	295	4	182
Motion, Mean Shift, Radial & Gaussian	0	295	4	182
Motion, Mean Shift, Radial & Gaussian (5 iterations each)	0	295	0	186

13

Challenges and Limitations

1. Dataset Suitability Challenges
2. Lack of Bounding Box Coordinates
3. Limited Recognition Capability
4. Complex Multi-Step Recognition Process

14

Conclusion

This project has illustrated how AI can enhance privacy because of automated face blurs. By using deep learning techniques and resolving some issues found in previous datasets, the present research sets a foundation for a safer and more secure online environment.

References

[1] S. S. Mohamed, W. A. Mohamed, A. T. Khalil, and A. S. Mohra, "Deep Learning Face Detection and Recognition," Intl Journal of Electronics and Telecommunications, pp. 1-8, 2024.

[2] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A Deep Learning Approach for Face Detection," 2018 IEEE Punecon, Pune, India, 2018, pp. 1-8. DOI: 10.1109/PUNECON.2018.8745376

[3] R. A. Vyas, "Feature Extraction Technique of PCA for Face Recognition With Accuracy Enhancement," International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol. 4, no. 11, pp. 288-291, Nov. 2016.

[4] M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," presented at the British Machine Vision Conference (BMVC), Swansea, UK, September 2015.

[5] T. Li and L. Lin, "AnonymousNet: Natural Face De-Identification with Measurable Privacy," presented at the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Thank You!!