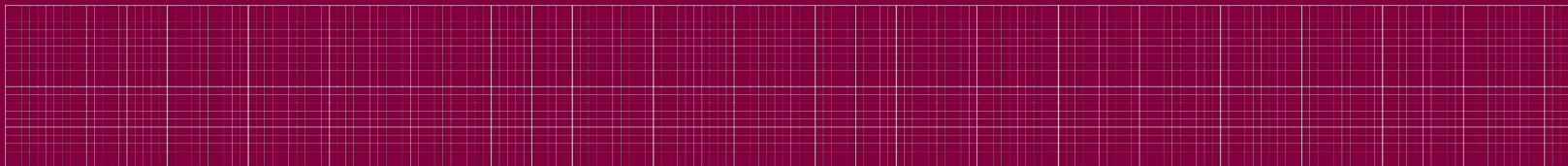




FACE PIXELATION: YOLO VS. MTCNN FOR FACIAL OBJECT DETECTION

COREY HUANG, ALLISON DEATON, KEN LIN, REBECCA YEUNG



0
1

INTRODUCTION

PROBLEM: FACE PIXELATION

We sacrifice more and more of our data every day on the internet. How can we maintain any privacy?

Face pixelation is a technique where one deliberately reduces the resolution of certain parts of an image to obscure one's face.

Where's the machine learning in face pixelation? It's in the actual face recognition!



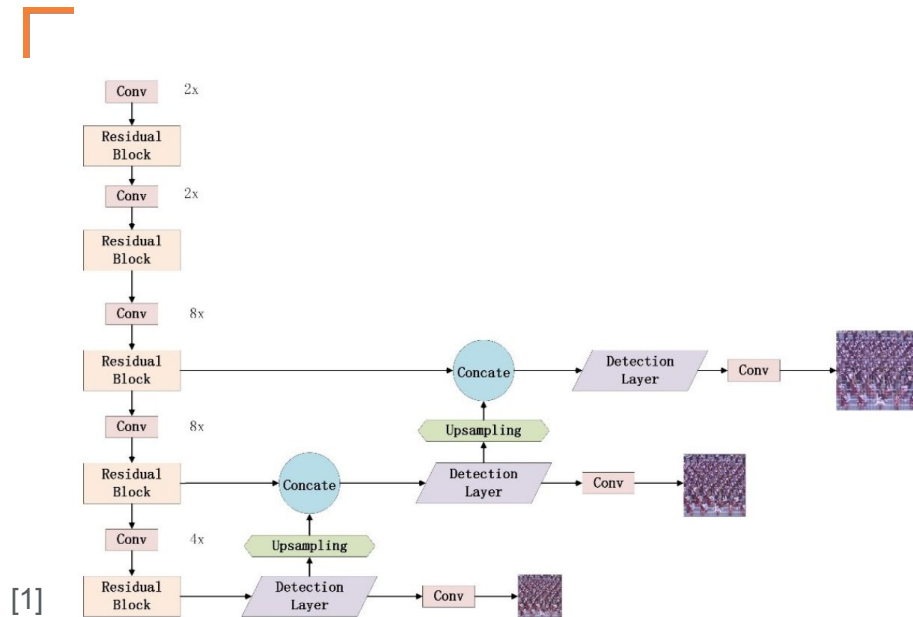
You Only Look Once (YOLO)

What is **YOLO**?

- ❑ A real-time object detection algorithm
- ❑ Single neural network predicts bounding boxes and class probabilities in a **single pass**.

Key features:

- ❑ Fast and efficient for real-time applications.
- ❑ Suitable for tasks like autonomous vehicles, surveillance, and robotics.



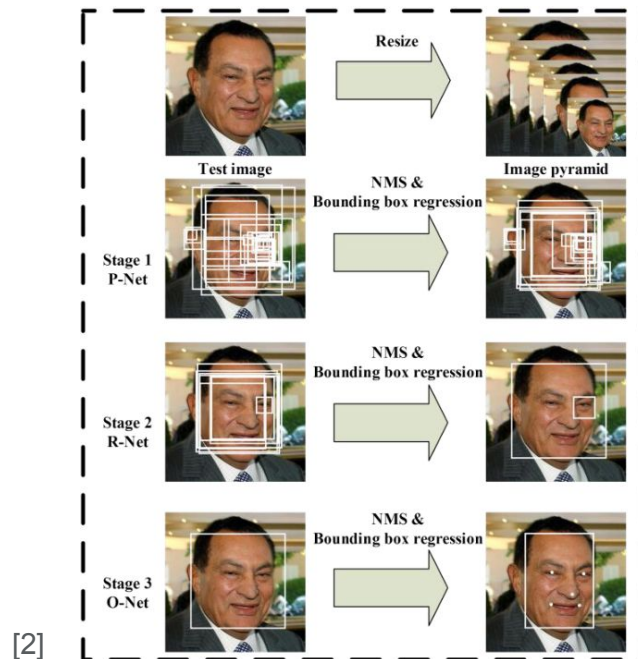
Multi-Task Cascaded Convolutional Networks (MTCNN)

What is **MTCNN**?

- ❑ A cascaded three-stage neural network to perform simultaneous face detection, bounding box prediction, and facial landmark localization.

How does it work?

- ❑ **P-Net**: Generates candidate face regions.
- ❑ **R-Net**: Refines and filters candidate regions
- ❑ **O-Net**: Outputs final bounding boxes and predicts facial landmarks



0
2

RELATED WORK

RELATED WORKS

- ❑ [Chang et al. \(2020\)](#) proposed an enhanced YOLO model for better performance on detecting tiny faces.
- ❑ [Tariyal et al. \(2024\)](#) conducted a comparative study of MTCNN, Viola-Jones, SSD and YOLO.
- ❑ [Yang et al. \(2016\)](#) introduced a massive dataset ideal for benchmarking face recognition, known as WIDERFACE.
- ❑ [Yu et al. \(2022\)](#) designed a model based on YOLO that performs better on face recognition tasks.
- ❑ [Zhang et al. \(2016\)](#) developed the initial MTCNN model containing multiple neural networks.
- ❑ [Zhang et al. \(2020\)](#) compared one-stage, two-stage, and three-stage (MTCNN) models for face recognition.
- ❑ [Zhang et al. \(2024\)](#) analyzed which facial features contribute the most to face recognition performance.

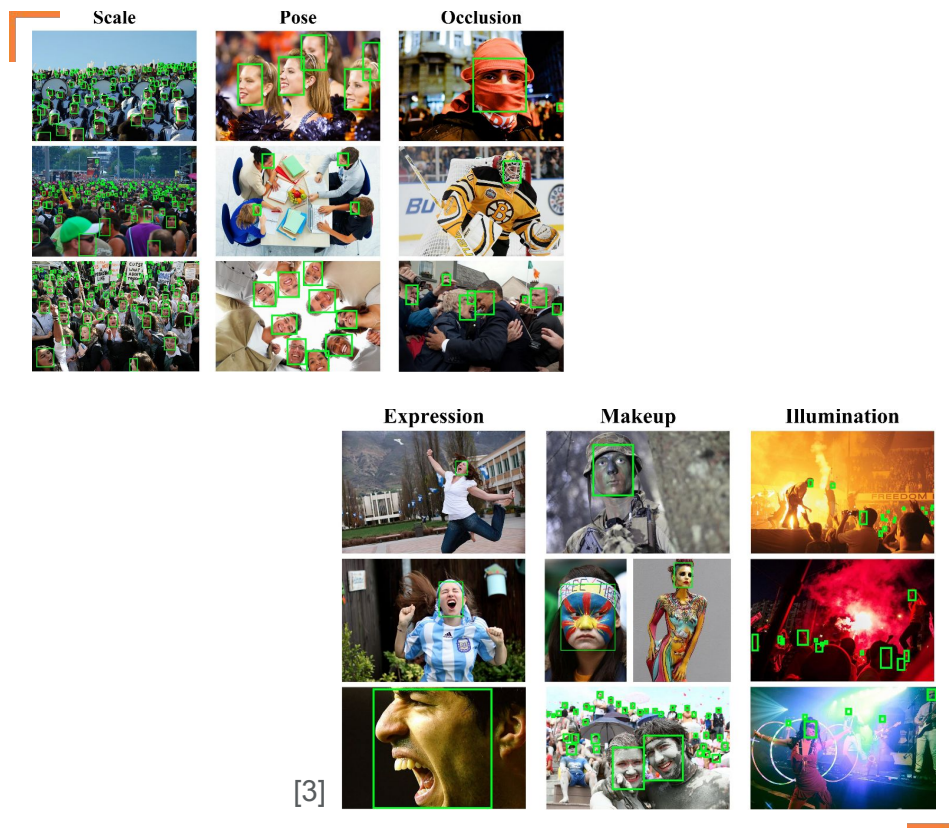
0
3

METHODOLOGY

DATASET: WIDERFACE

WIDERFACE is a face detection benchmark dataset based on a subset of images from the publicly available **WIDER** dataset.

- ❑ 32,203 images
- ❑ 393,703 labeled faces



SAMPLE SIZE

Random sample taken from each dataset.

We used smaller samples for speed.

Train (~70%):

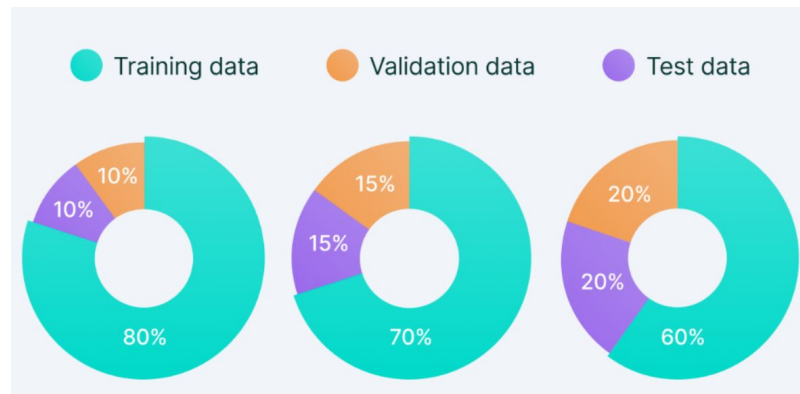
- ❑ 500 images

Validation (~15%):

- ❑ 100 images

Test (~15%):

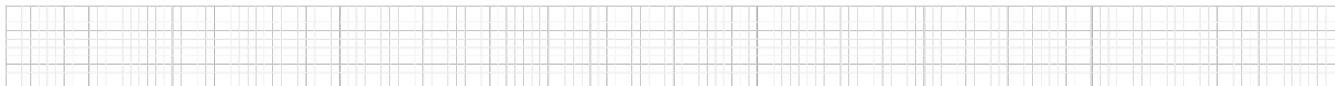
- ❑ 100 images



INFRASTRUCTURE

The code was run using **Google Colab**'s virtual machine system. We used their allocated GPU system when running our YOLO algorithm analysis.

Colab allocates a **Tesla T4 GPU** for free account usage – roughly equivalent to a Nvidia GeForce RTX 2070 Super.



IMPLEMENTATION

Annotations:

- ❑ Object class (face)
- ❑ Bounding box coordinates
- ❑ Converted for YOLO for train and validation sets

Train:

- ❑ Trained model using YOLOv8 with train set
- ❑ Evaluated with validation set

Test:

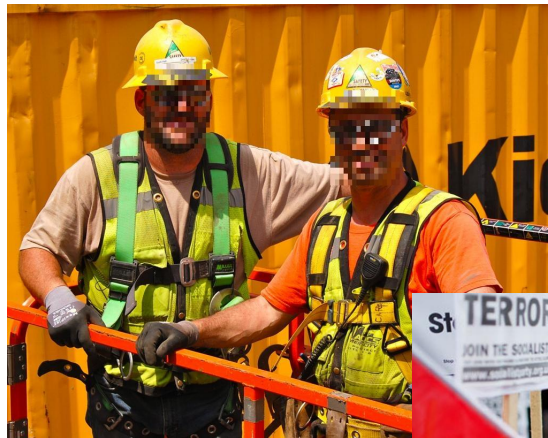
- ❑ Predict and apply bounding boxes to test set



IMPLEMENTATION

Pixelation:

- ❑ Get labels with coordinates provided from testing
- ❑ Apply pixelation to predicted bounding box area
- ❑ Reduce face region to 16x16 pixels using linear interpolation
- ❑ Resize back to original using nearest-neighbor interpolation



HYPERPARAMETERS

Our implementation uses these default hyperparameters for training.

Image Size:

- ❑ Resolution for training and validation (default 640)

Epochs:

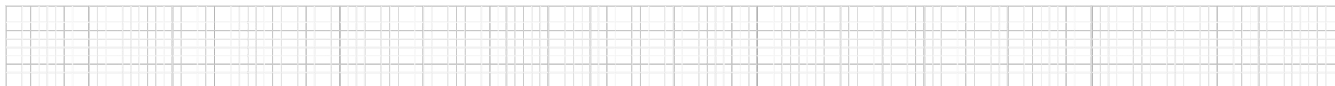
- ❑ Number of full passes through dataset (default 50)

Batch Size:

- ❑ Number of images processed together in one iteration (default 16)

Workers:

- ❑ Number of CPU threads to load data (default 4)



DIMENSIONS OF ANALYSIS

What do we want to find out?

Accuracy:

- ❑ How close are the bounding boxes to their true locations?

Adaptability:

- ❑ Different features and/or occlusion

Speed:

- ❑ Speed vs. Accuracy tradeoff

0
4

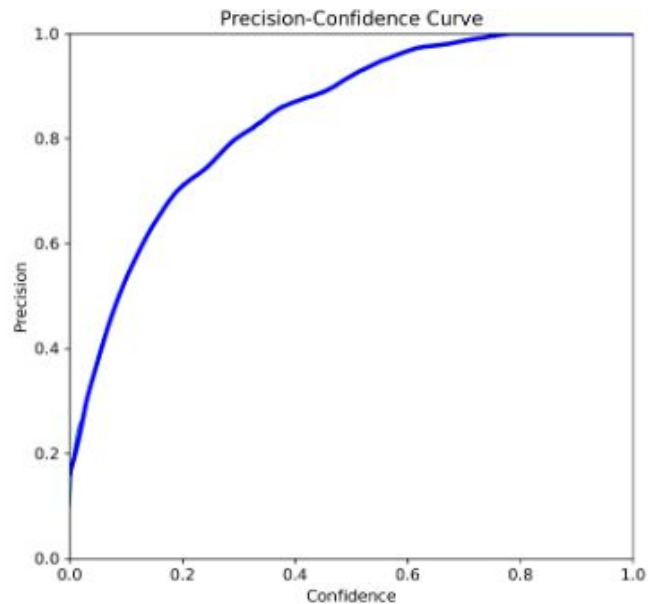
ANALYSIS

PRECISION

Out of all predictions that were positive, how many were actually correct?

As confidence increases, less positive predictions are accepted. Therefore, fewer false positives means higher precision.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$



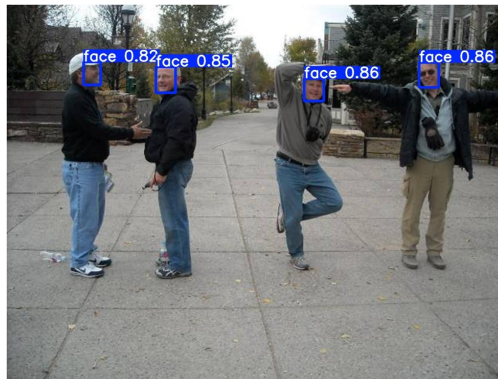
PRECISION

High Precision:

- ❑ Often correct when predicting positive
- ❑ Detects face when it's there

Low Precision:

- ❑ Many false positives
- ❑ Detects face when there actually isn't one



HIGH PRECISION



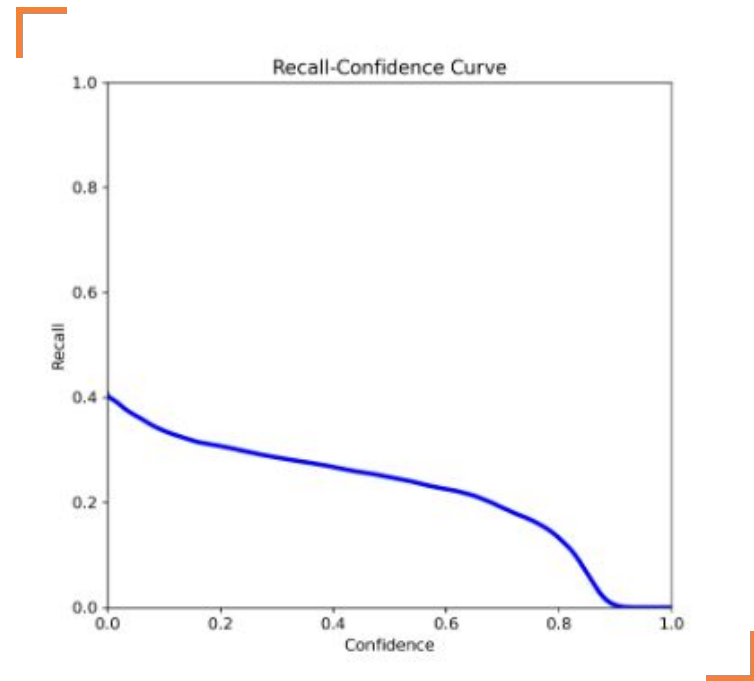
LOW PRECISION

RECALL

Out of all actual positives, how many were successfully predicted?

As confidence decreases, more positive predictions are accepted. Therefore, more true positives means higher recall.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$



RECALL

High Recall:

- ❑ Predicts most of true positives
- ❑ Detects most faces in image

Low Recall:

- ❑ Many false negatives
- ❑ Did not detect faces that were there



HIGH RECALL

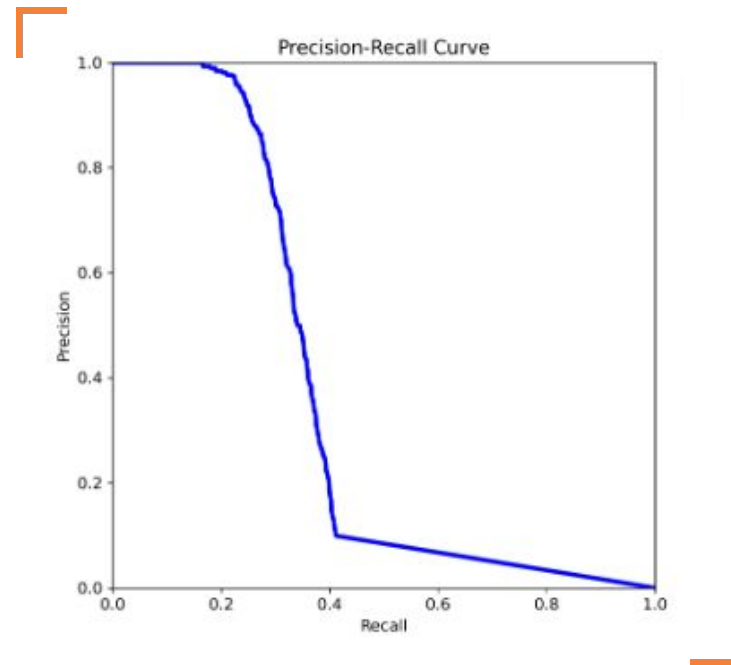


LOW RECALL

PRECISION-RECALL TRADEOFF

Going for high precision will minimize false positives, but true positives can also be missed which decreases recall.

Going for high recall will increase true positives but also false positives which can reduce precision.

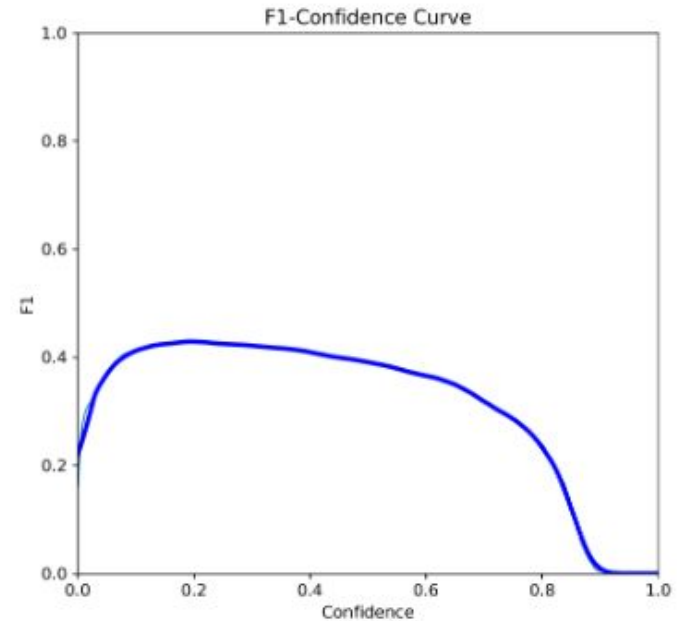


F1 SCORE

Shows balance between precision and recall to evaluate performance.

Determines which confidence threshold is optimal.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



ACCURACY

mAP50-95:



- ❑ Measures accuracy of predictions
- ❑ Mean of AP values over 10 IoU thresholds (0.50 to 0.95 in steps of 0.05)

Mean Average Precision (mAP):

- ❑ Average Precision (AP) is found in area under Precision-Recall Curve

Intersection over Union (IoU):

- ❑ Measures overlap between predicted and true bounding box
- ❑ Higher IoU means predicted box is closer to ground truth

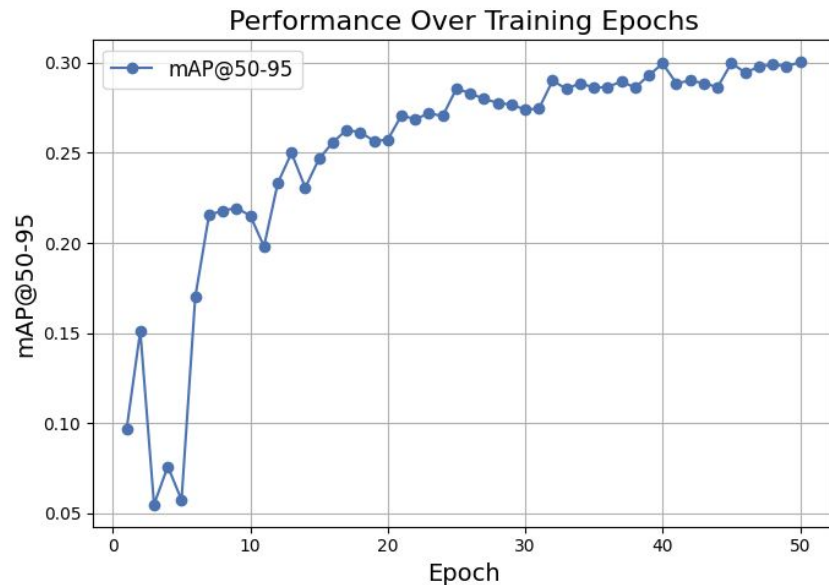

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


PERFORMANCE

Good range for mAP50-95 is 0.3 - 0.4.

Improving Performance:

- ❑ Increase sample size
- ❑ More epochs
- ❑ Lower learning rate (better generalization)
- ❑ Lower batch size (more updates = better generalization but worse gradients)
- ❑ Higher batch size (less updates = better gradients but worse generalization)



SPEED

CPU vs. GPU:

- ❑ CUDA for GPU acceleration (parallel processing)

Model Architecture Size:

- ❑ YOLOv8 comes in different sizes (we are using smaller model)
- ❑ Smaller models increase speed but lower accuracy (fewer layers)

Batch Size:

- ❑ Process more data for each iteration, reducing number needed for each epoch

Image Size:

- ❑ Smaller images but can reduce detection accuracy

YOLO VS MTCNN

Metric	MTCNN	YOLO
Speed	Slower due to 3-stage pipeline	Fast, single-pass, real-time detection
Accuracy	High specifically for faces and landmarks	General-purpose detection
Landmark Detection	Built-in	Non-existent
Applications	Detection under difficult circumstances including shifting lighting, changing poses, and occlusions	Live video analysis, video conferencing and monitoring

0
5

LIMITATIONS

LIMITATIONS

This study has only performed numerical analysis as detailed in section 04 on a face pixelation process that uses the YOLOv8 algorithm for face recognition.

Currently, any comparisons to MTCNN are primarily qualitative and based on existing literature in the field.

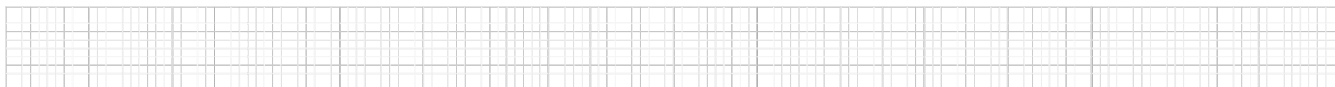
0
6

FUTURE WORK

CONCLUSION

Using YOLO, we were able to detect faces, evaluate performance through mAP and other metrics, and analyze methods to improve our model.

In general, YOLO lends itself to real-time applications where speed is a necessity. MTCNN's multistage process is more accurate under difficult conditions, and can also identify facial landmarks.



























FUTURE WORK

One area of future work could be running a set of tests on both YOLO and MTCNN with a focus on **occluding** facial features.

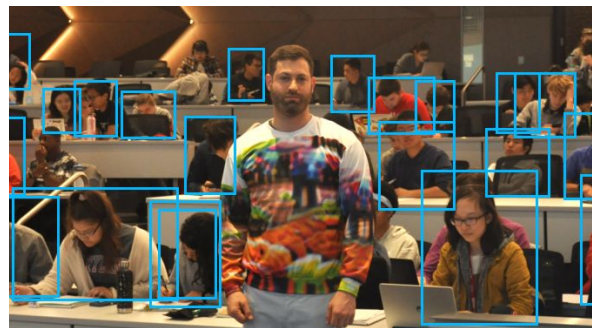
- ❑ How easy is it to protect one's privacy just by using a mask or covering your eyebrows with a hat?

Another area could be exploring how the different face recognition algorithms work when faced with **adversarial** patterns.

- ❑ Could adversarial makeup have merit?

	Origin	Level_1	Level_2	Level_3	Level_4	Mask
Eyebrow						
Eyes						
Mouth						
Nose						

[4]



[5]

0
7

REFERENCES

[1] Jia-Yi Chang, Yan-Feng Lu, Ya-Jun Liu, Bo Zhou, and Hong Qiao. 2020. Long-distance tiny face detection based on enhanced yolov3 for unmanned system.

[2] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks.

[3] “WIDER FACE: A Face Detection Benchmark,” Shuoyang1213.me, 2017. <http://shuoyang1213.me/WIDERFACE/>

[4] Qianqian Zhang, Yueyi Zhang, Ning Liu, and Xiaoyan Sun. 2024. Understanding of facial features in face perception: insights from deep convolutional neural networks.

[5] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. 2020. Making an invisibility cloak: real world adversarial attacks on object detectors. (2020).

