Face Pixelation: YOLO vs. MTCNN for facial object detection

Corey Huang
Virignia Tech
Blacksburg, VA, USA
chuang16@vt.edu

Allison Deaton
Virginia Tech
Blacksburg, VA, USA
deatona404@vt.edu

Ken Lin
Virignia Tech
Blacksburg, VA, USA
lken@vt.edu

Rebecca Yeung
Virginia Tech
Blacksburg, VA, USA
ryeung@vt.edu

Abstract

The increasing demand for privacy on the Internet has led to growing interest in face pixelization, a process involving the detection and obfuscation of faces in images. This project explores the implementation and evaluation of modern facial detection algorithms to facilitate effective face pixelization. Specifically, we focus on the YOLO (You Only Look Once) algorithm, a convolutional neural network-based object detection framework, and compare its performance with other established algorithms like MTCNN (Multi-Task Convolutional Neural Network).

In this paper, we evaluate the YOLO algorithm using the WIDER-FACE dataset for training, validation, and testing. This study aims to provide insights into the practical performance of YOLO for face pixelization and compare it with the performance of MTCNN.

ACM Reference Format:
Corey Huang, Allison Deaton, Ken Lin, and Rebecca Yeung. 2024. Face Pixelation: YOLO vs. MTCNN for facial object detection. CS 5805, Blacksburg, VA, USA, 5 pages.

1 Introduction

This paper specifically seeks to explore two major machine learning algorithms used in facial detection and pixelization: YOLO and MTCNN. These two algorithms have gained prominence for their unique and efficient approaches to face detection.

MTCNN uses a cascaded framework of convolutional neural networks to detect faces to quickly and accurately detect faces in complex scenes. The MTCNN model consists of three layers of convolutional neural networks, called the P-Net (Proposal Network), R-Net (Refine Network), and O-Net (Output Network), which operate in a cascade to detect faces in an image. The first network (P-Net) generates the candidate bounding boxes. These are then refined by the second network (R-Net). Finally, the third neural network (O-Net) refines these bounding boxes further while simultaneously predicting facial landmarks such as eyes, nose, and mouth positions. This allows the MTCNN model to detect faces with high precision in a variety of scenarios where the image might not show a face under ideal conditions, such as occulsion, varying lights, and complex backgrounds.

YOLO is a single-stage object detector, which has more speed and accuracy than its two-stage counterparts and are more widely used in applications. It uses a single neural network to predict bounding boxes and probabilities in a single pass, making it an incredibly quick method for facial detection tasks. This algorithm divides the image into a grid and predicts the bounding boxes and probabilities for each grid cell in a single forward pass through a neural network. This makes YOLO very suitable for real-time applications, where speed and efficiency are key, such as video surveillance systems. Recent versions, such as YOLOv5 and YOLOv6, have further refined its performance, introducing advancements in model architecture and training techniques to improve accuracy and robustness.

This paper aims to compare the YOLO and the MTCNN algorithms in the context of facial recognition to evaluate their performance in terms of accuracy, adaptability, and speed. By analyzing these metrics, we hope to provide practical insights into their suitability under a variety of less-than-ideal conditions.

2 Literature Review

Upon deciding on our broader project topic, that of facial pixelization, we came to the conclusion that the primary concern in such processes is the actual facial detection subprocess. Once the pixels representing a face are identified, most algorithms to blur, pixelize, or otherwise obfuscate high-resolution images are quite simple and plainly mathematical. With that in mind, we focused our research on machine learning algorithms used in facial detection.

2.1 You Only Look Once

Our group immediately jumped to the YOLO (You Only Look Once) family of object detection algorithms, as multiple of our members have worked with it in the past. Analyzing its efficacy in comparison to algorithms traditionally used for primarily facial object detection could prove interesting. So, we first decided to search for any existing research involving YOLO models and facial recognition, and found a few good background papers from recent years.

YOLOv3 inherently uses multi-scale detection, that is, the ability of a model to identify objects of varying sizes within an image by analyzing features at different scales. This algorithm can be further refined using feature pyramids and feature reorganization to create YOLOv3-C. These techniques enhance the detection of small,

CS 5805, December 2024, Blacksburg, VA, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

medium, and large objects (in this case, faces) by improving the way features are combined across different layers of the neural network with the goal of extracting more detailed and discriminative information about facial features [1]. This is key for detecting small or distant faces, such as when they are in a large, crowded environment. Although we are not implementing the YOLOv3-C algorithm described here, we will be comparing the speed and accuracy of a few YOLO models with MTCNN in our analysis.

Many real-time face detectors have been developed based on various YOLO models. For instance, in 2022, Ziping Yu et al. introduced YOLO-FaceV2 [6], a model built on YOLOv5. YOLOv5, an enhanced version of YOLOv3, features a more advanced algorithm for efficient feature extraction, achieving higher performance and a smaller model size [2]. Leveraging these advancements, YOLO-FaceV2 further improves the detection of small objects, addressing challenges faced by earlier models. The key contribution here is the introduction of the Receptive Field Enhancement (RFE) module for improved multi-scale feature recognition [6]. RFE uses multiple branches to analyze the image with varying dilation rates. Each branch looks at the image at a different scale to enhance the network's ability to understand features that are either unusually large or very small.

With each iteration of YOLO, the algorithm has demonstrated significant advancements in both efficiency and accuracy. Each of these models has evolved to address specific challenges in object detection, such as improving the detection of small and distant objects, handling scaled variations, and optimizing computation speed for use in real-time applications. This has made YOLO an incredibly versatile tool for a wide range of applications, including facial detection, where precision and speed can be critical.

2.2 Multi-Task Cascaded Convolutional Networks

After a decent bit of research into YOLO's usage in face detection, we then started researching other methods of facial detection to contrast it to. Older algorithms like Haar cascade classifiers were commonly mentioned in research papers and machine learning articles, but comparing a modern algorithm like YOLO to classifiers used decades ago didn't seem particularly fair. Another algorithm that was frequently mentioned was MTCNN (Multi-task convolution neural network). This algorithm was first proposed in a 2016 article and, since it was a far more recent development than other facial detection algorithms we were researching, we picked it as our second algorithm to analyze. We took a look at the original proposal article, and a more recent analysis of its accuracy from 2020.

The multi-task CNN is exactly what it says on the tin: a cascade of different CNNs perform different tasks to better isolate and identify faces. The first stage uses a shallow CNN to quickly identify "candidate windows" that are then refined in a second stage to reject non-face candidates, using a more complex CNN. Finally, that information is passed through a third CNN to further refine the results and identify the actual facial landmarks [7]. Using multiple CNNs in this manner allows the performance to be optimized on a by-CNN, by-task basis.

One issue that can arise through this multistage process is the development of overlapping cross-sections (visibly, the bounding boxes) for the same actual target face. A solution to this is adding a non-maximum suppression (NMS) algorithm to the process, which selects the highest confidence window and removes overlapping windows based on their IOU (intersection over union) value—essentially, a mathematical representation of how much they overlap with our ideal target window [7] [8].

The MTCNN typically treats the face classification stage (is there a face in the section of the image we are checking?) as a binary classification process, and uses a cross-entropy loss function for training. For the task of candidate frame regression, a difference-square-loss function is used, as the bounding box data is a (x,y,w,h) vector and therefore a regression task [8]. Because of its multistage process, the MTCNN algorithm foregoes speed and capability for real-time applications as a trade-off for higher accuracy and ability to detect faces in more situations (such as crowds, when occluded, etc.).

2.3 Possible Dimensions of Analysis

Next, we decided to solidify what properties we wanted to analyze. Obviously, accuracy is a main thing to compare, but what exact measures of accuracy? Should we aim for detecting individual facial features, or just the existence of a face in general? Are we looking for a binary correctness on if a face is discovered, or how close a predicted bounding box is to its true location? Many of these dimensions were things we could decide on based on general prior knowledge or trends from previous papers, but we also sought out comparative research that could help us identify further dimensions to analyze.

One area of interest could be analyzing how the different algorithms react to the occlusion of certain facial features, such as blurring or occluding one or both eyes, one or both eyebrows, the nose, or the mouth. Prior research has found that facial detection systems rely heavily on the detection of eyebrows as a key feature in detection, so occluding those in particular may be particularly insightful [9]. We could also examine how the algorithms treat faces with subtle differences, such as if they find it easier to identify male versus female faces, or if they struggle to identify faces with certain expressions. We could also try blurring out certain features with a Gaussian blur rather than fully occluding them, or introducing camera artifacts like noise, as these sometimes produce different results from occlusion [9]. Finally, we could measure facial recognition accuracy when a large number of face objects are present (such as within a crowd) between the different algorithms.

We are also considering a significant analysis of the speed of our algorithms. In real-world applications, it is quite common to need to trade-off in favor of speed rather than precision/accuracy [3]. It would be appropriate for us to measure the speed at which the algorithms recognize a face or faces in sets of images. We could also see if any of the facial occlusion metrics have any impact on speed, or if they only impact accuracy.

A few of the papers also mentioned dataset size as a variable of interest in face recognition algorithm analysis. We sought out a large dataset that we could potentially cut into smaller subsets to test how the algorithms would function if trained on smaller datasets. Seeing if one of the algorithms is particularly resistant to a small training dataset could prove beneficial for real-world applications.

We ultimately decided on the WIDERFACE dataset due to its extensive coverage of the features we aimed to focus on, namely detecting faces in complex and diverse conditions. This data set contains abundant annotations for a variety of scenarios, including occlusion, pose, and event categories [5]. These characteristics align closely with our objective of evaluating face detection models under real-world conditions. Using this comprehensive dataset allows us to ensure that our analysis captures a wide range of facial attributes and environments to effectively assess our chosen models.

3 Analysis

3.1 Precision, Recall, and F1-Score

In the end, we decided on a straightforward set of analysis dimensions. Our first dimension was precision: or, out of all predictions that were positive, how many were actually correct?

We consider true positives as the cases where the model correctly detects a face and the detected bounding box sufficiently overlaps with a ground truth bounding box. False positives are the cases where the model predicts a bounding box as a face, but there was either no corresponding ground truth (a face was detected where there is no face), or an insufficient overlap between the bounding box and the ground truth. False negatives are the cases where the model fails to detect a face that is present in the ground truth. True negatives are the cases where the model correctly identifies that there is no face in any particular region.

In other words, no bounding box is predicted in areas where there is no ground truth face. Then, precision is the ratio of true positives to the total number of faces detected by the model (true positives and false positives). A high precision indicates that the model is often correct in its predictions, meaning that most of the detected faces are actually true faces as per the ground truth. A low Precision shows that the model predicts many false positives.

$$Precision = \frac{true\ positives}{true\ positives\ +\ false\ positives}$$

Our next dimension was recall, which measures the model's ability to identify all the faces present in the ground truth. Out of all actual positives, how many were successfully predicted? Recall is the ratio of true positives to the total number of positives (true positives and false negatives). As confidence decreases, more positive predictions are accepted. A high recall suggests that the model can successfully detect most faces, while a low recall indicates that the model is unable to identify many faces that are there in the ground truth.

$$Recall = \frac{true\ positives}{true\ positives\ +\ false\ negatives}$$

The F1-Score is the harmonic mean of Precision and Recall. This value shows balance between precision and recall to evaluate performance. It determines which confidence threshold is optimal by identifying the threshold that provides the best trade-off between Precision and Recall. In YOLO, the confidence threshold determines the minimum confidence level at which the model considers a prediction valid.

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

3.2 YOLO Numerical Analysis

For our analysis of YOLO, we chose to use the WIDERFACE face detection benchmark dataset. WIDERFACE is based on a subset of images from the publicly available WIDER dataset, and contains a multitude of different face-containing images with a wide variety of positions, scales, expressions, and levels of occlusion [5]. We used smaller samples from this dataset for speed. Our training subset was about 70% of the sample (500 images), our validation subset was about 15% (100 images) and our testing subset was also about 15% (100 images).

For training, we used an image resolution of 640 pixels, 50 epochs, a batch size of 16 images processed together in one iteration, and a total of 4 CPU threads working concurrently on loading data. Our analysis was done on the YOLOv8 model.

After using YOLO to identify face coordinates, we implemented a pixelation process on the detected faces. The face region was reduced to 16x16 pixels using linear interpolation, and was resized back to the original size using jagged, nearest-neighbor interpolation (Figure 1).

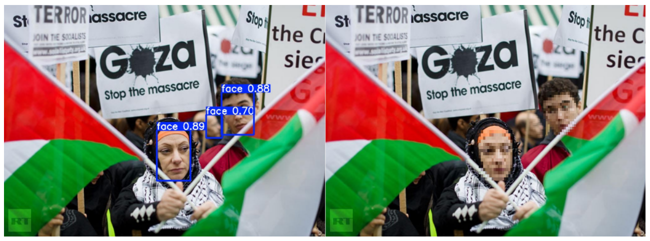

Figure 1: Comparison of facial recognition (left) with confidence scores and anonymization (right) using pixelation.

The YOLO model had a positive correlation between object detection confidence and precision. The relation was a concave-down curve where the precision increased sharply at confidence levels below 0.5, and had a smaller slope for confidence levels between 0.5 and 1.0 (Figure 2). If we set a low confidence threshold, the model will produce more predictions (higher coverage) at the cost of lower precision. A higher confidence threshold will make fewer predictions with higher precisions, albeit at the potential cost of missing a few bounding boxes that should have been predicted.

For recall, the model displayed a negative correlation between recall and confidence. Recall sharply fell off with confidence threshold values above 0.8 (Figure 2). It appears as though the YOLO model rarely detects face objects with higher confidence values such as those above 0.8. A lower confidence threshold is required to avoid false negatives in face recognition when using the YOLO model.

The F1 Score metric, which shows a balance between precision and recall, ended up as a concave-down curve with a local and absolute maximum of 0.2 (Figure 2). This means that for the YOLO model,

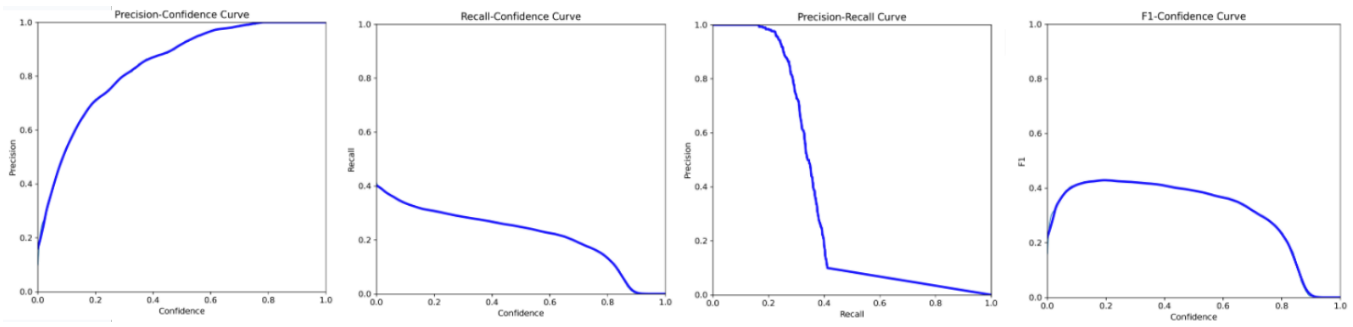

Figure 2: Graphs showing precision vs. confidence, recall vs. confidence, precision vs. recall, and F1-score vs. confidence

that would be a good confidence threshold to use for determining if a face object is detected or not.

The mean average precision (mAP50-95) measures the overall accuracy of predictions by calculating the average precision (AP) at different IOU thresholds. This evaluation considers IoU thresholds ranging from 0.50 to 0.95, in increments of 0.05. By averaging multiple IoU thresholds, this metric ensures a robust evaluation of the model's ability to make accurate and tight bounding box predictions. A higher IoU indicates that the predicted box is closer to the ground truth than otherwise.

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

According to our analysis, a good range for the mAP50-95 is 0.3 - 0.4. Figure 3 plots mAP50-95 against training epochs. During the early training phase, the mAP increases rapidly, indicating that the model is learning effectively in the initial epochs. In later epochs, the mAP50-95 plateaus around 0.3, showing that the model is converging and that any further improvement would be minimal.

There are several ways to boost performance. The easiest method would be to increase the sample size. Adding more labeled data can help the model learn better representations and generalize to unseen data. We could also train for more epochs, which would help if the model hasn't converged yet. A lower learning rate would allow for more precise updates to the model weights. A lower batch size processes less images together which allows for more updates (better generalization) but each update can have more noisy data because there are less images at once. However, higher batch size processes more images together which has better gradients per update but there are less updates overall.

Unfortunately, YOLO works best with a sample size in the thousands. However, considering our time constraints and limited processing power, we had to shrink the sample size we used down to a few hundreds. Nonetheless, we still have satisfactory results.

3.3 YOLO and MTCNN

In terms of speed, MTCNN is slower overall due to its 3-stage pipeline, which processes images in multiple steps to refine predictions (e.g. bounding boxes and facial landmarks). On the other hand, YOLO is extremely fast due to its single-pass architecture. It is able to perform detection in a single network pass. This makes YOLO ideal for real-time uses cases where getting results as quickly

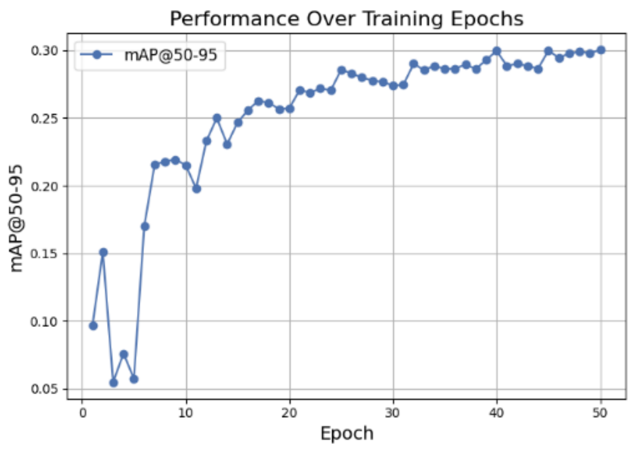

Figure 3: Graph showing mAP@50-95 score for each training epoch

as possible is critical, while MTCNN would be more suitable for applications where speed is less important.

Additionally, MTCNN is designed specifically for facial detection and landmark localization. It achieves high accuracy in detecting faces under challenging conditions, such as varying lighting, occulsions, and different poses. As such, MTCNN has built-in landmark detection. YOLO is more of a general-purpose detection models, with the ability to detect a broad range of objects (not just faces). While it can be accurate for many things, it may not perform as well as MTCNN for specialized tasks like face detection or fine-grained landmark localization. Since it is meant to be for general object detection, YOLO does not have any default landmark detection.

In terms of application, MTCNN is best used for face-specific tasks, such as detection under difficult circumstances that include changing lighting, changing poses, and occlusions. YOLO would ideally be used for live video analysis, video conferencing and monitoring.

4 Conclusion

Both YOLO and MTCNN are fast, modern algorithms that are very capable of recognizing faces with a high precision. However, they both fill their own niche: MTCNN, through its multiple networks of P-net, R-net, and O-net, all of which can be fine-tuned, often succeeds in situations where YOLO fails. However, due to that complexity, it takes multiple passes, and is somewhat slower than YOLO, making it less suitable for real-time application. YOLO provides an excellent precision-speed tradeoff, with an acceptable accuracy in one pass, making it the preferred choice for real-time usage.

While detecting faces to anonymize them with pixelation is one method of preserving biometric privacy, recent researchers have proposed surveillance-obstructive fashion as another method. Clothing such as t-shirts, sweaters, or pants that contain adversarial patterns can act as a modern anti-computer "invisibility cloak" to mitigate recognizing humans in a typical object detection task [4]. Attempting to expand upon this topic and developing and evaluating methods of creating facial recognition invisibility cloaks could be an avenue for future work in the cross-section of data privacy and machine learning.

Another idea for future work is a more comprehensive numerical analysis of how both YOLO and MTCNN perform when given images with significant occlusion. Occluding or covering certain facial features such as eyebrows can have a significant performance impact on face recognition algorithms [9]. A case-by-case comparison of the same images on the different object detection algorithms under those conditions could prove insightful. How much does hiding the lower half of a face with a face mask impact face detection? Does it only impact confidence levels, or does it significantly impact the accuracy of the bounding box? Are certain algorithms more adaptable to these changes in face recognizability? These sorts of analysis questions are far more relevant to our initial inspiration of facial data privacy and its protection.

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. *CoRR*, abs/2010.04421. https://arxiv.org/abs/2010.04421 arXiv: 2010.04421.

[2] Aicha Khalfaoui, Abdelmajid Badri, and Ilham EL Mourabit. 2022. Comparative study of yolov3 and yolov5's performances for real-time person detection. In *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, 1–5. doi: 10.1109/IRASET52964.2022.9737924.

[3] Sumit Tariyal, Rahul Chauhan, Yogesh Bijalwan, Ruchira Rawat, and Rupesh Gupta. 2024. A comparitive study of mtcnn, viola-jones, ssd and yolo face detection algorithms. In *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 1–7. doi: 10.1109/IITCEE59897.2024.10467445.

[4] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. 2020. Making an invisibility cloak: real world adversarial attacks on object detectors. (2020). https://arxiv.org/abs/1910.14667 arXiv: 1910.14667 [cs.CV].

[5] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. Wider face: a face detection benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5525–5533. doi: 10.1109/CVPR.2016.596.

[6] Ziping Yu, Hongbo Huang, Weijun Chen, Yongxin Su, Yahui Liu, and Xiuying Wang. 2022. Yolo-facev2: a scale and occlusion aware face detector. (2022). https://arxiv.org/abs/2208.02019 arXiv: 2208.02019 [cs.CV].

[7] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. In number 10. Vol. 23. Institute of Electrical and Electronics Engineers (IEEE), (Oct. 2016), 1499–1503. doi: 10.1109/lsp.2016.2603342.

[8] Ning Zhang, Junmin Luo, and Wuqi Gao. 2020. Research on face detection technology based on mtcnn. In *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. doi: 10.1109/ICCNEA50255.2020.00040.

[9] Qianqian Zhang, Yueyi Zhang, Ning Liu, and Xiaoyan Sun. 2024. Understanding of facial features in face perception: insights from deep convolutional neural networks. In vol. 18. doi: 10.3389/fncom.2024.1209082.