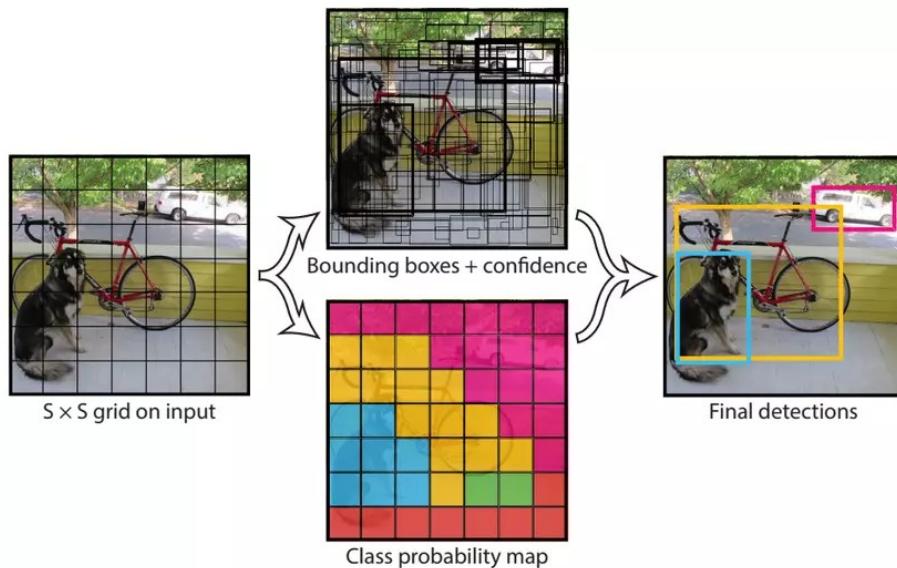


YOLO - You only look once

1. Face Detection

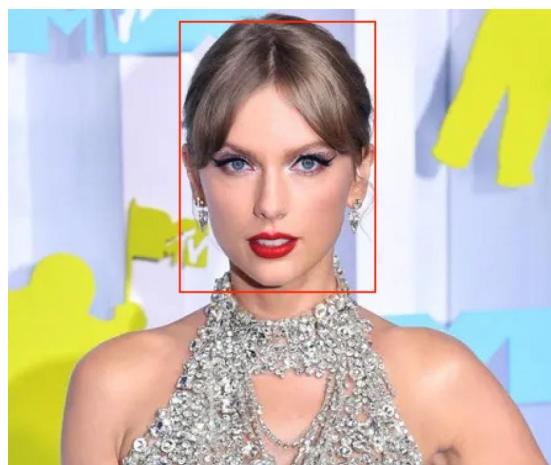
Đồ án thực hiện tác vụ Face Detection dùng Yolov7

a. Tổng quan của mô hình Yolo Object Detection



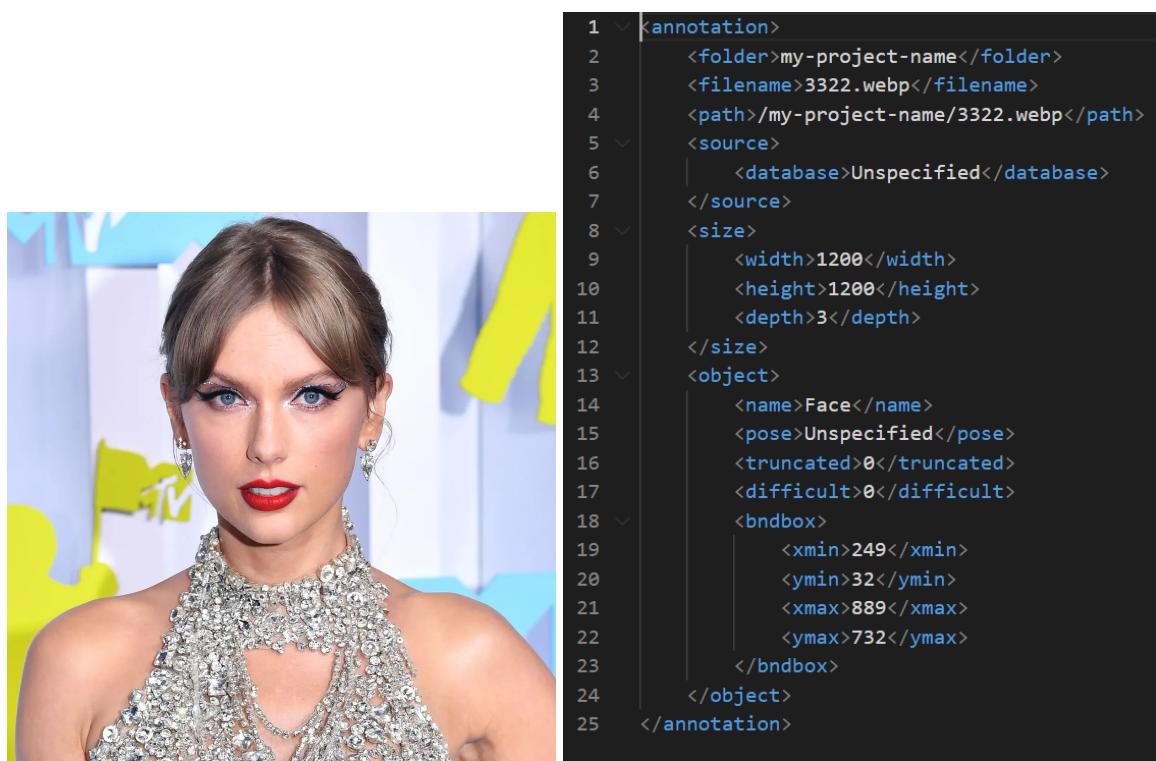
Hình: Kiến trúc tổng quan của Yolo cho bài toán object detection

b. Dữ liệu training

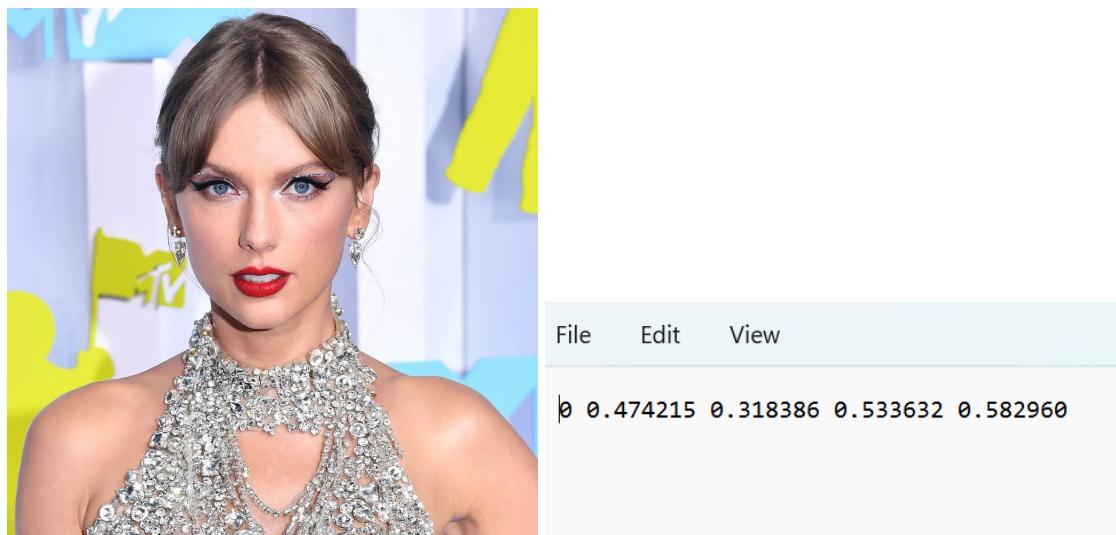


Hình: Dữ liệu hình ảnh được đánh bounding box

Dữ liệu được dùng để huấn luyện cho các mô hình Yolo có thể là các chú thích bounding box viết bằng xml hoặc text file, với mỗi object mang thông tin về loại object, vị trí và độ lớn của bounding box.

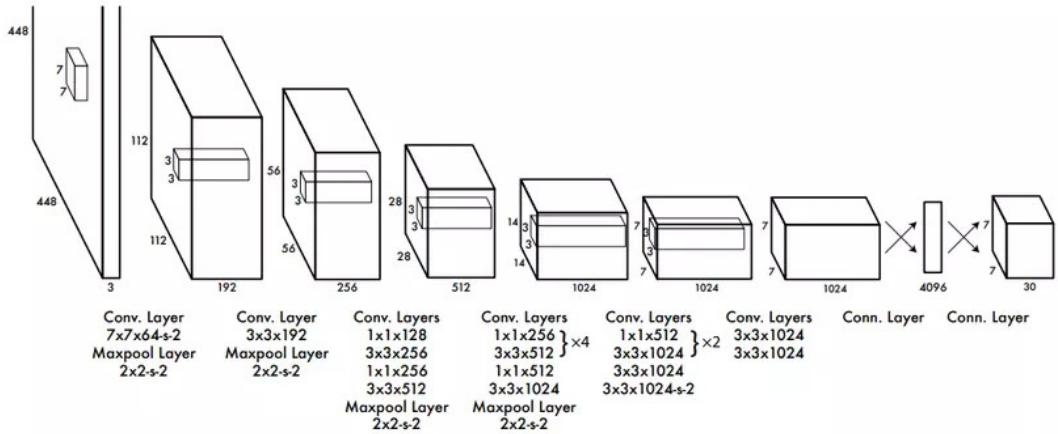


Hình: Data hình ảnh và chú thích bằng file xml



Hình: Data hình ảnh và chú thích bằng file text

c. Yolo basic structure



Hình: Kiến trúc cơ bản của Yolo Object detection

Yolo với backbone là CNN với nhiều lớp convolution chồng lên nhau, có các neck kết hợp các kết quả lại với nhau và head để dự đoán kết quả phân lớp của object.

Trường hợp ảnh được chia thành 7x7 grid cells, mỗi grid cell góp vào dự đoán cho 2 boundingbox và gồm 20 class predictions.

d. Yolo output

Trong 30 giá trị của mỗi grid cell bao gồm:

- 2 bounding box information
- c value: có chứa object không
- x: tọa độ tâm bounding box theo phương x
- y: tọa độ tâm bounding box theo phương y
- w: chiều rộng bounding box
- h: chiều cao bounding box
- 20 giá trị xác suất của ô đó chứa object class 1, class 2, ...class 20

Vì vậy $2*5+10=30$

→ Mỗi cell dự đoán được 1 object (bounding box có xác suất chứa cao nhất), điều này tạo nên bất lợi của Yolo là không thể xác định nhiều object trùng lặp trong 1 cell.

Prediction score:

$$\text{Pr}(\text{Class}_i | \text{Object}) * \text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \text{Pr}(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Mỗi grid cell sẽ nhận được prediction score của từng lớp. Tại một thời điểm, Yolo chỉ muốn mỗi bounding box thì “chịu” trách nhiệm cho duy nhất một object.

e. Yolo object detection

Dự báo bounding box:

Các giá trị chỉ vị trí được chỉnh về [0,1] để bounding box không lọt ra ngoài ảnh (sử dụng các hàm regression,...)

Lượt bỏ số lượng khung hình dùng non-max suppression:

Bước 1: giảm số lượng bounding box bằng cách lọc bỏ bounding box có xác suất chứa vật thể thấp (ví dụ threshold_1=0.5)

Bước 2: đối với các bounding box giao nhau, lấy bounding box có xác suất chứa vật lớn nhất, tính các IoU với các bounding box còn lại, nếu IoU cao hơn threshold_2 thì 2 bounding box đang giao nhau rất cao, có thể loại bỏ.

f. Yolo loss function

Classification loss: Tương tự như sum-square error cho classification cho tất cả các grid cell.

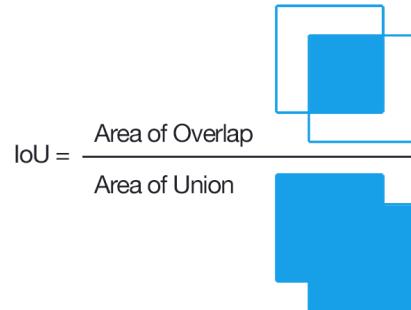
$$L_{classification} = \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2$$

Trong đó:

\mathbb{I}_i^{obj} : bằng 1 nếu ô vuông đang xét có object ngược lại bằng 0

$\hat{p}_i(c)$: là xác suất có điều của lớp c tại ô vuông tương ứng mà mô hình dự đoán

IOU: Giá trị ngưỡng 0 đến 1, đo sự overlap giữa kết quả dự đoán và ground truth. là độ đo sự chính xác quan trọng trong các tác vụ có sử dụng chú thích IoU càng cao chứng minh càng có sự overlap lên nhau



Localization loss:

- Bởi vì 1 grid cell chỉ dự đoán phân lớp cho 1 object và cell's predictor mà chịu trách nhiệm dự báo cho lớp đó sẽ là cell có được IOU hiện tại cao nhất với ground truth.
- Độ lỗi tính chênh lệch khoảng cách giữa bounding box thật sự và kết quả prediction.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

Tương tự khi tính độ lỗi của tọa độ, tuy nhiên chiều rộng và cao của bounding box được lấy căn bậc 2 ý muốn thể hiện: các thay đổi nhỏ của các box lớn thì không quan trọng bằng các sự sai lệch trong các box nhỏ.

Confidence loss:

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

- Xét cho tất cả các Bounding Box và tất cả các grid cell.
- C là confidence score của grid cell i, C_hat là IOU của predicted bounding box với ground truth.
- 1 obj =1 khi có object trong cell 0 khi không có và 1 noobj thì ngược lại.

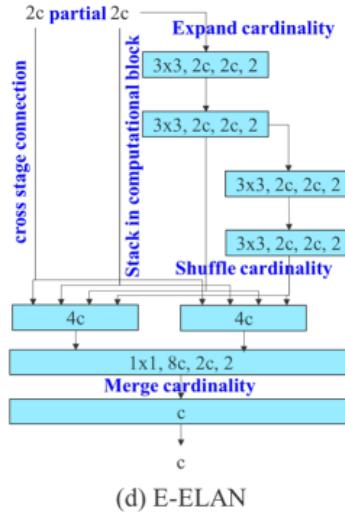
Total loss: Được tính bằng tổng của cả 3 loss thành phần.

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

g. Yolov7

Những cải tiến của Yolov7 so với các phiên bản trước đó:

- Kết hợp các lớp mở rộng hiệu quả: không thay đổi kiến trúc ban đầu, tận dụng các nhóm tích chập: sử dụng hiệu quả các tham số, tính toán.



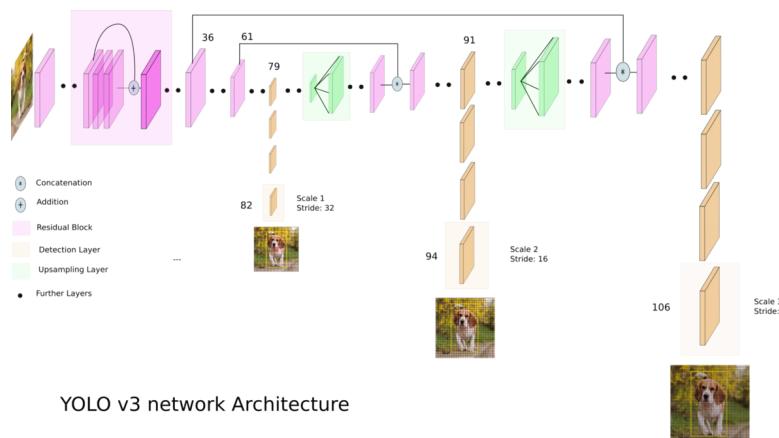
(d) E-ELAN

Hình: Extended efficient layer aggregation networks

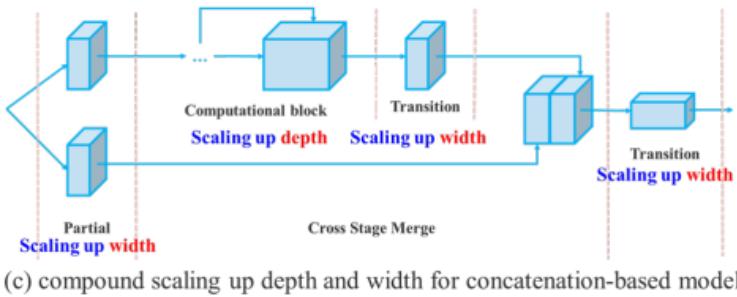
ELAN mở rộng (E-ELAN) hoàn toàn không thay đổi đường truyền gradient của kiến trúc ban đầu, nhưng sử dụng tích chập nhóm để tăng số lượng tính năng của các tính năng được thêm vào và kết hợp các tính năng của các nhóm khác nhau theo cách xáo trộn và hợp nhất số lượng tính năng.

- Model Scaling:

Yolov7 thực hiện chia tỷ lệ độ sâu và rộng khác nhau rồi concat các layer này lại với nhau giữ cho kiến trúc mô hình tối ưu trong khi mở rộng quy mô cho các kích thước khác nhau.

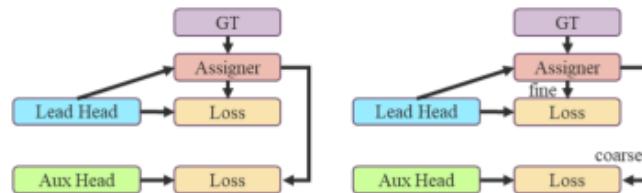


Hình: Kiến trúc của Yolov3

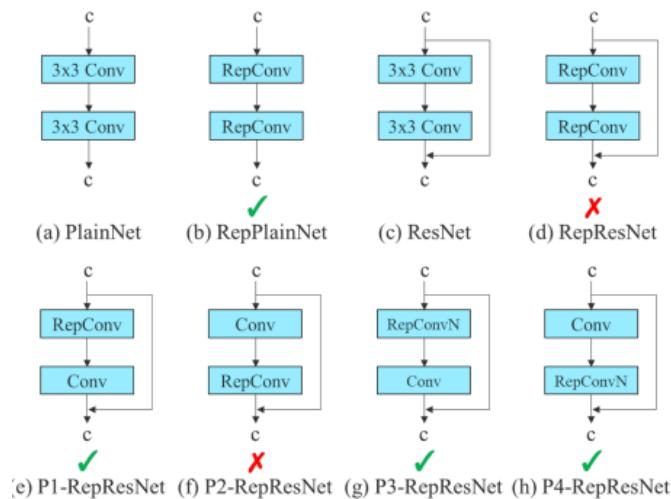


Hình: Model scaling của Yolov7

- Auxiliary head: Yolov7 thêm các head vào ở khoảng giữa các tầng mạng nhằm “giám sát” kết quả phân lớp, mục đích dẫn kết quả về lớp đúng của nó



- Ứng dụng kỹ thuật tái tham số hóa: lấy trung bình một tập hợp các trọng số của mô hình để tạo ra một mô hình mạnh mẽ hơn so với các mẫu chung mà nó đang cố gắng lập mô hình.



Hình: Planned re-parameterized model.

RepConv là lớp kết hợp giữa conv 3x3, conv 1x1 và identity connection thành 1 conv layer

h. Yolov5Face

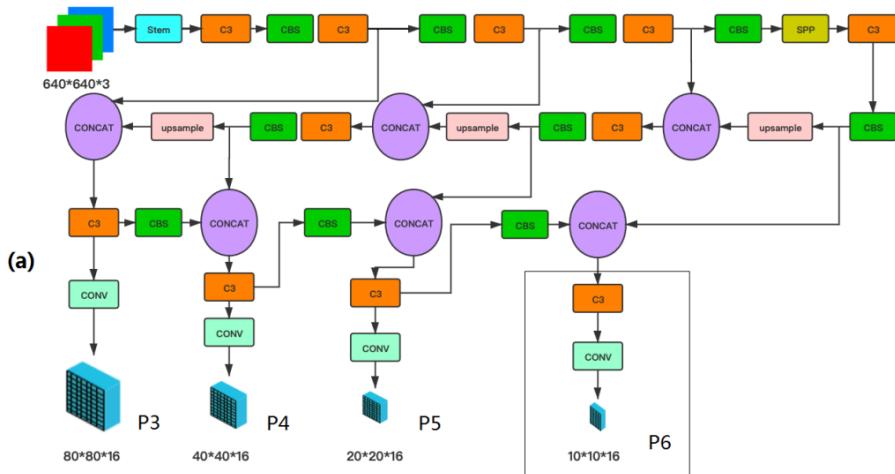
Tại sao phải sử dụng YoloFace:

- Được train riêng trên tập face dataset, như vậy sẽ học được đặc trưng mặt người và phân biệt mặt người tốt hơn.



Hình: Wider Face dataset

- Yolov5 được implement thành backbone và cải thiện để optimize cho tác vụ face detection. Các thông số, kiến trúc mô hình được tinh chỉnh chỉ dành cho tác vụ face detection.



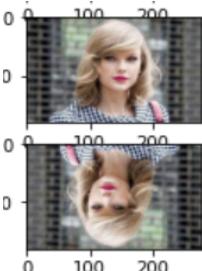
Hình: Kiến trúc tổng quan của Yolov5Face

Focus layer of YOLOv5 được thêm bằng Stem block structure giúp tăng khả năng tổng quát hóa của mô hình đồng thời giảm độ phức tạp tính toán.

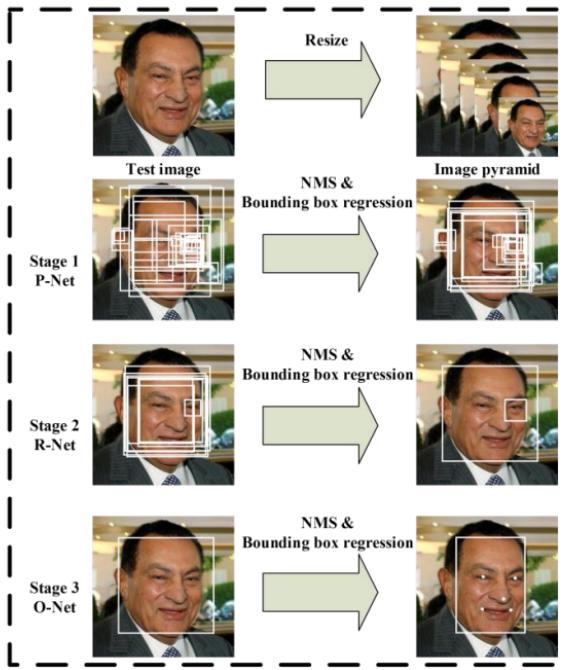
SPP được thay bằng kernel nhỏ hơn, giúp YOLOv5 phù hợp hơn với face detection và tăng accuracy.

- Một số tác vụ augmentation khi thực nghiệm không phù hợp với tác vụ face detection cũng bị thay đổi. Ví dụ: up-down flipping and Mosaic.

Hình: Flip up-down không phù hợp cho tác vụ detect mặt người



- Landmark - điểm mốc là những điểm quan trọng trên khuôn mặt con người và được thêm vào thành các head của YoloFace như vậy ảnh sẽ được cẩn chỉnh khuôn mặt trước khi đưa đi nhận dạng.



Hình: MTCNN network tạo ra kết quả các landmark trên mặt người
Output label của YoloFace được thêm trực tiếp 5 điểm landmarks.

- Sử dụng hàm wing loss:

$$wing(x) = \begin{cases} w \cdot \ln(1 + |x|/e), & \text{if } x < w \\ |x| - C, & \text{otherwise} \end{cases} \quad (1)$$

w không âm đặt phạm vi của phần phi tuyến thành $(-w, w)$, e giới hạn độ cong của vùng phi tuyến và $C = w - w\ln(1 + w/e)$ là hằng số liên kết “smooth” các phần- các phần tuyến tính và phi tuyến xác định (khả vi).

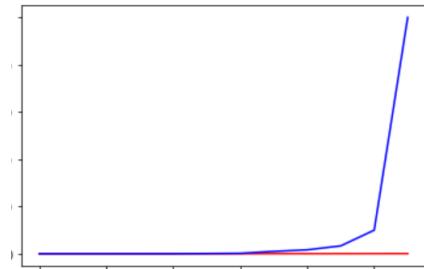
The loss functions for landmark point vector $s = \{s_i\}$, and its ground truth $s' = \{s'_i\}$, where $i = 1, 2, \dots, 10$, is defined as,

$$loss_L(s) = \sum_i wing(s_i - s'_i) \quad (2)$$

Wing loss sensitive với các error nhỏ vì thế thích hợp cho tác vụ detect mặt người vì trong quá trình train dữ có sự khác nhau rất nhỏ giữa các khuôn mặt làm cho mô hình nhạy cảm hơn với việc phát hiện object là người.

Thực nghiệm Với x là line màu xanh, line màu đỏ tương ứng với $\ln(1+x/e)$.

```
x=[20.0,10.0,5,1.0,0.5,0.1,0.05,0.01,0.006,0.003,0.001,0.0001]
```



Total loss: Với hàm loss cho general object là $loss_O$ (bounding box, class, probability) thì hàm loss của YoloFace:

$$loss(s) = loss_O + \lambda_L \cdot loss_L$$

Reference

- [1] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv preprint arXiv:2207.02696 (2022).
- [2] Qi, Delong, Weijun Tan, Qi Yao, and Jingfeng Liu. "YOLO5Face: why reinventing a face detector." arXiv preprint arXiv:2105.12931 (2021).
- [3] <https://blog.roboflow.com/yolov7-breakdown/>
Nguồn tư liệu hình ảnh được lấy từ Internet