

Identifying ID Submissions using Computer Vision

Improving the customer experience in applying for a new digital bank account



Contents

01

Introduction

02

Exploratory Data Analysis

03

Modelling & Deployment

04

Conclusion





Introduction



Indonesia has a Huge Untapped Banking Market, but ID Verification is Laborious & Time-Consuming

Scenario: A digital bank in Singapore wants to expand its banking business into the Indonesian market

Opportunity:

- **Large under-served market:** 4th most populous nation, 3rd largest unbanked population¹
- **Young, tech-savvy & tech-hungry population:** median age is 30 years; has the 2nd highest interest level in digital financial services (Southeast Asia)²

Challenge:

- **No national digital ID:** for account opening, laborious & time-consuming ID verification must be done with digital submission of physical documents (i.e. photos)

50%

(~140m) of population
yet to have a bank account¹

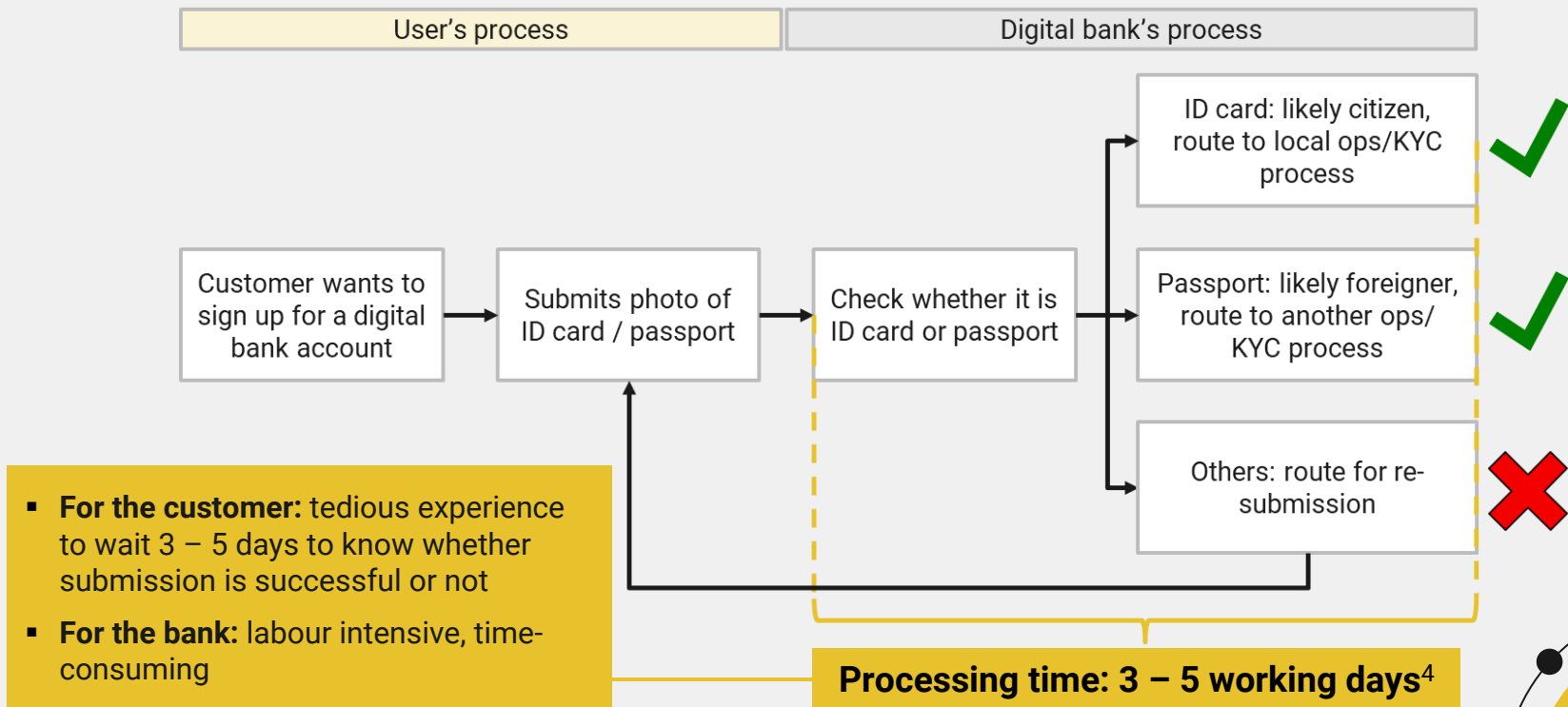
146m

people are between
15 – 49 years of age³



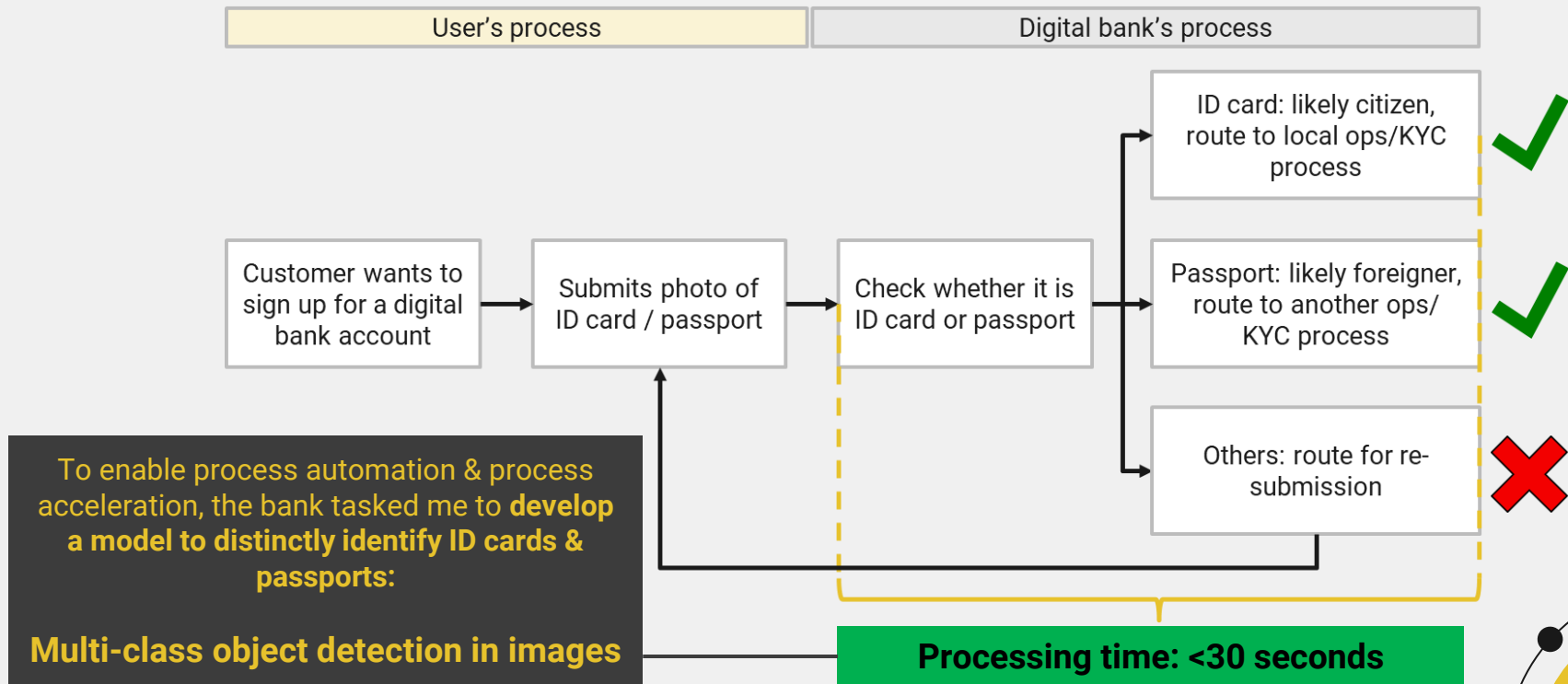
Problem Statement: manual verification of ID submissions results in poor customer experience

Currently, ID Verification takes: 3 – 5 Working Days



[4] as reference: an incorrect photo submission to Singapore's ICA may double the process time (The Straits Times, May 22): 2 weeks for NRIC, 4 weeks for passport (ICA, n.d.)

Goal: Reduce to <30 Seconds for Each Submission



Approach: To Train YOLOv5 Model on ID Documents

01

Label & Analyse Dataset

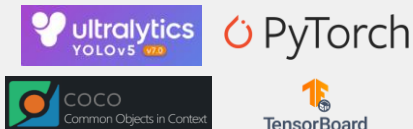
As proof-of-concept, to use **artificially-generated ID cards and passports** from East Europe, North-West Asia



02

Model Training

Use **YOLOv5** for its state-of-the-art inference speed, good accuracy performance & ease of use



03

Model Deployment via Streamlit

Use **Streamlit** to illustrate **model prediction speed & capability** – users can choose from a selected set of photos, or upload their own





Exploratory Data Analysis (EDA)



Dataset: 1K photos from East Europe & North-West Asia⁵

| | Train 600 photos | Validation 200 photos | Test 200 photos |
|-----------|--|---------------------------------|---------------------------|
| ID cards | 300 from Slovakia, Spain, Finland (100 each) | 100 from Estonia | 100 from Albania |
| Passports | 300 from Latvia, Russia, Greece (100 each) | 100 from Serbia | 100 from Azerbaijan |

- **Mock ID documents:** from East Europe and North-West Asia, each with unique text fields and artificially generated faces
- **Separated nationalities:** Train, validation & test datasets each have unique nationalities to prevent data leakages
- **Resized:** all photos were resized to 640 x 640 pixels for YOLO

[5] More details in [Annex](#)

Dataset: Illustration* (1)

Normal



Different background



Different brightness



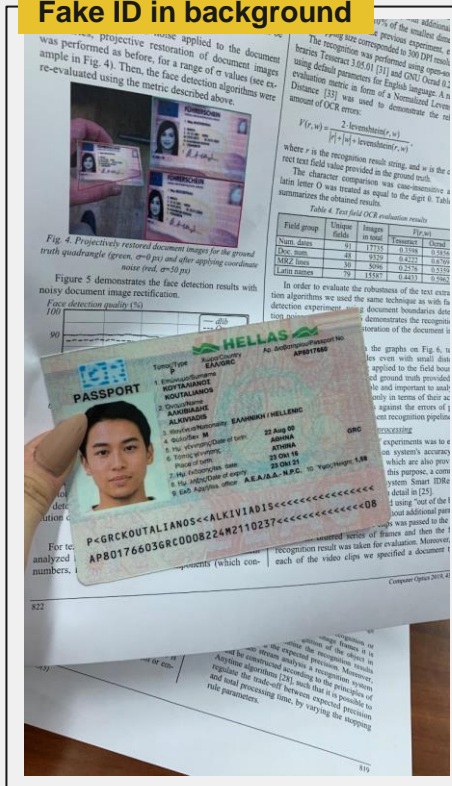
*Illustrated before photo resize to 640 x 640 pixels

Dataset: Illustration (2)

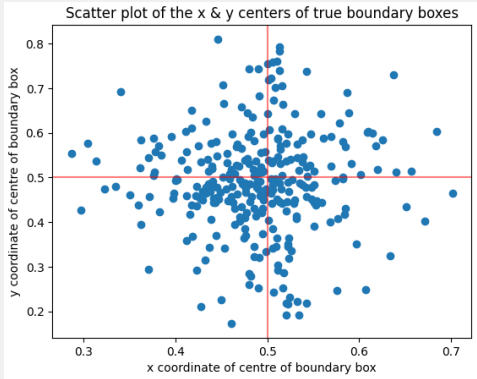
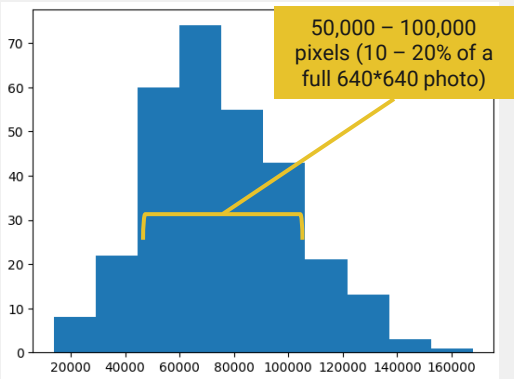
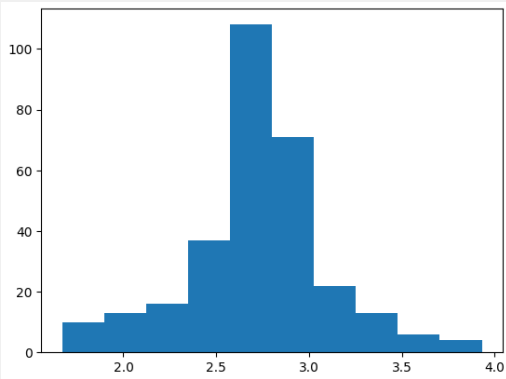
Multiple cards



Fake ID in background

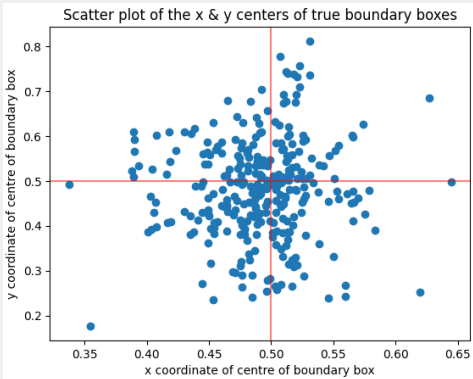
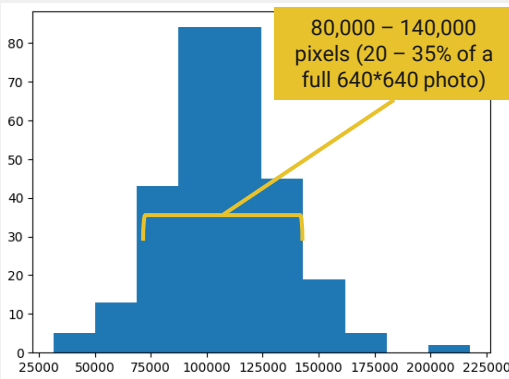
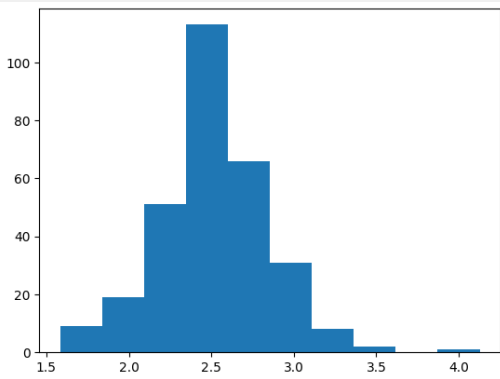


EDA: Training ID Cards are Usually in the Center, 10-20% of Full Photo Size, have Standard ID Aspect Ratio

| | Scatterplot of Bounding Box centres | Histogram of Bounding Box Area | Histogram of Bounding Box Aspect Ratio |
|----------|--|---|--|
| Graphs |  |  |  |
| Analysis | ID cards usually in centre of photo & along the red lines → trained model may be weak in identifying cards in photo corners | No samples near 409,600 pixels (640 * 640) → trained model may be weak in identifying close-up photos of ID cards | Most hover between 2.5 – 3.0 (standard is 2.6 for an ID card*) → trained model may be weak in identifying rotated ID cards |

*ID card's standard aspect ratio of 2.6 is after resizing photos to 640 * 640 pixels

EDA : Training Passports are Similar to ID Cards

| | Scatterplot of Bounding Box centres | Histogram of Bounding Box Area | Histogram of Bounding Box Aspect Ratio |
|----------|--|--|---|
| Graphs |  |  |  |
| Analysis | <p>Passports usually in centre of photo & along the vertical red line → trained model may be weak in identifying passports in left, right & photo corners</p> | <p>No samples near 409,600 pixels (640 * 640) → trained model may be weak in identifying close-up photos of passports</p> | <p>Most hover around 2.5 (standard is 2.6 for an passport*) → trained model may be weak in identifying rotated passports</p> |

*Passport's standard aspect ratio of 2.6 is after resizing photos to 640 * 640 pixels

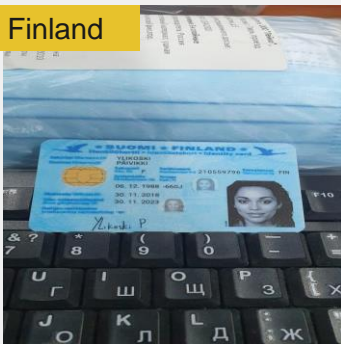
EDA : Training ID Cards, Passports' Colour Themes

ID Cards:
**Blue,
Yellow**

Spain



Finland



Slovakia

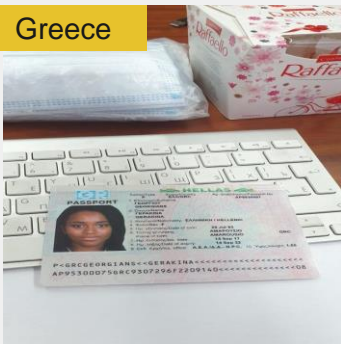


Passports:
White, Pink

Latvia



Greece



Russia



EDA Takeaway: to explore **data augmentation** for training dataset to increase sample variations



Modelling & Deployment



Modelling Overview

| | Baseline Model | Improved Model 1 | Improved Model 2 | Improved Model 3 |
|-------------------------------|--|--|--|--|
| Dataset | 1000 photos (600 train, 200 validation, 200 test) | | | 1600 photos (1200 train, 200 validation, 200 test) |
| Augmentation on training data | None | <ul style="list-style-type: none">▪ Colours: hue, saturation, brightness⁶▪ Image translation, scale, flip left-right, mosaic⁷ | <ul style="list-style-type: none">▪ Colours: hue, saturation, brightness▪ Image translation, scale, flip left-right, mosaic⁷▪ Rotation, shear⁷ | <ul style="list-style-type: none">▪ Colours: hue, saturation, brightness▪ Image translation, scale, flip left-right, mosaic⁷ |
| Model training, validation | YOLOv5 Large^{8,9}: <ul style="list-style-type: none">▪ Batch size = 8 (limited by hardware¹⁰)▪ Epochs = 100▪ Starting weights = pretrained weights on COCO 2017 | | | |
| Model testing | <ul style="list-style-type: none">▪ Best weight from training-validation▪ Confidence threshold = 0.7▪ With test-time augmentation | | | |

[6] Detailed explanation of colour augmentation in [annex](#)

[7] Detailed explanation of dimensional augmentation in [annex](#)

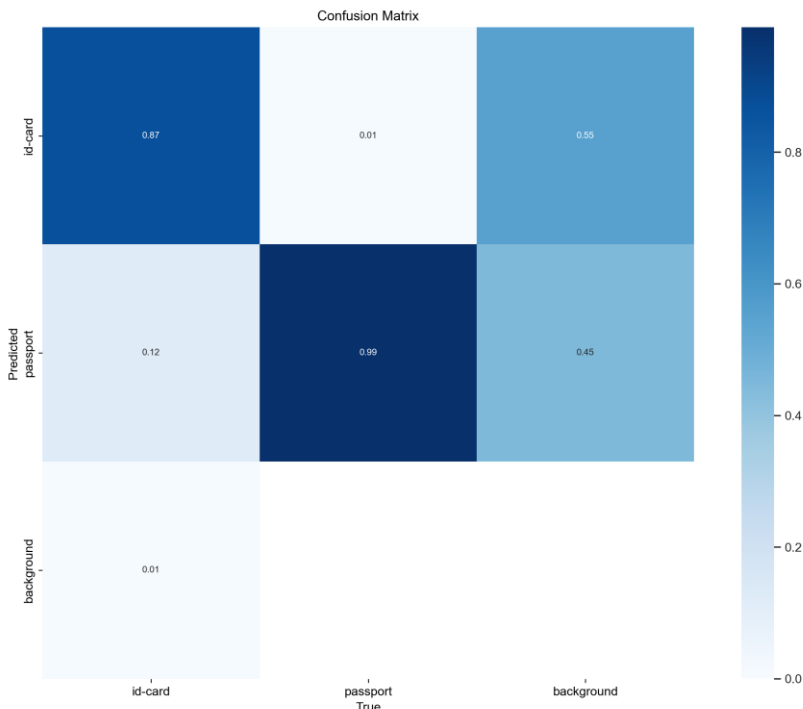
[8] YOLO performance over other models in [annex](#), YOLOv5l performance over other variants in [annex](#)

[9] Reference for training parameters for small dataset by [Ultralytics, May 2022](#)

[10] Hardware used: Intel i5-12400F, RAM 16GB, Nvidia RTX 3060

Baseline Model

Validation Scores



Test Scores

| | ID Cards | Passport |
|---------------------------------------|----------|----------|
| Correctly identified | 99 | 62 |
| Wrongly identified as the other class | 1 | 25 |
| Wrongly identified as background | 0 | 13 |

Analysis for Train-Val:

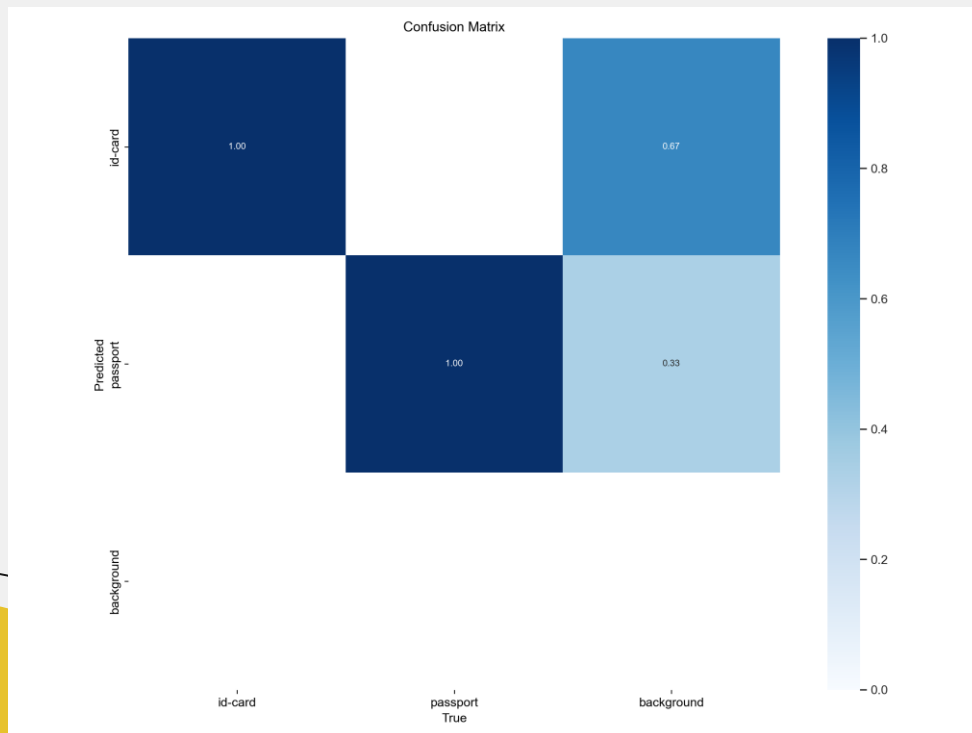
- Did well for passport, did decently for ID card
- Identified some background as ID card/Passport

Analysis for Test:

- Performed poorly for passport prediction

Improved Model 1 (+ Augmentation)

Validation Scores



Test Scores

| | ID Cards | Passport |
|---------------------------------------|----------|----------|
| Correctly identified | 95 | 99 |
| Wrongly identified as the other class | 5 | 1 |
| Wrongly identified as background | 0 | 0 |

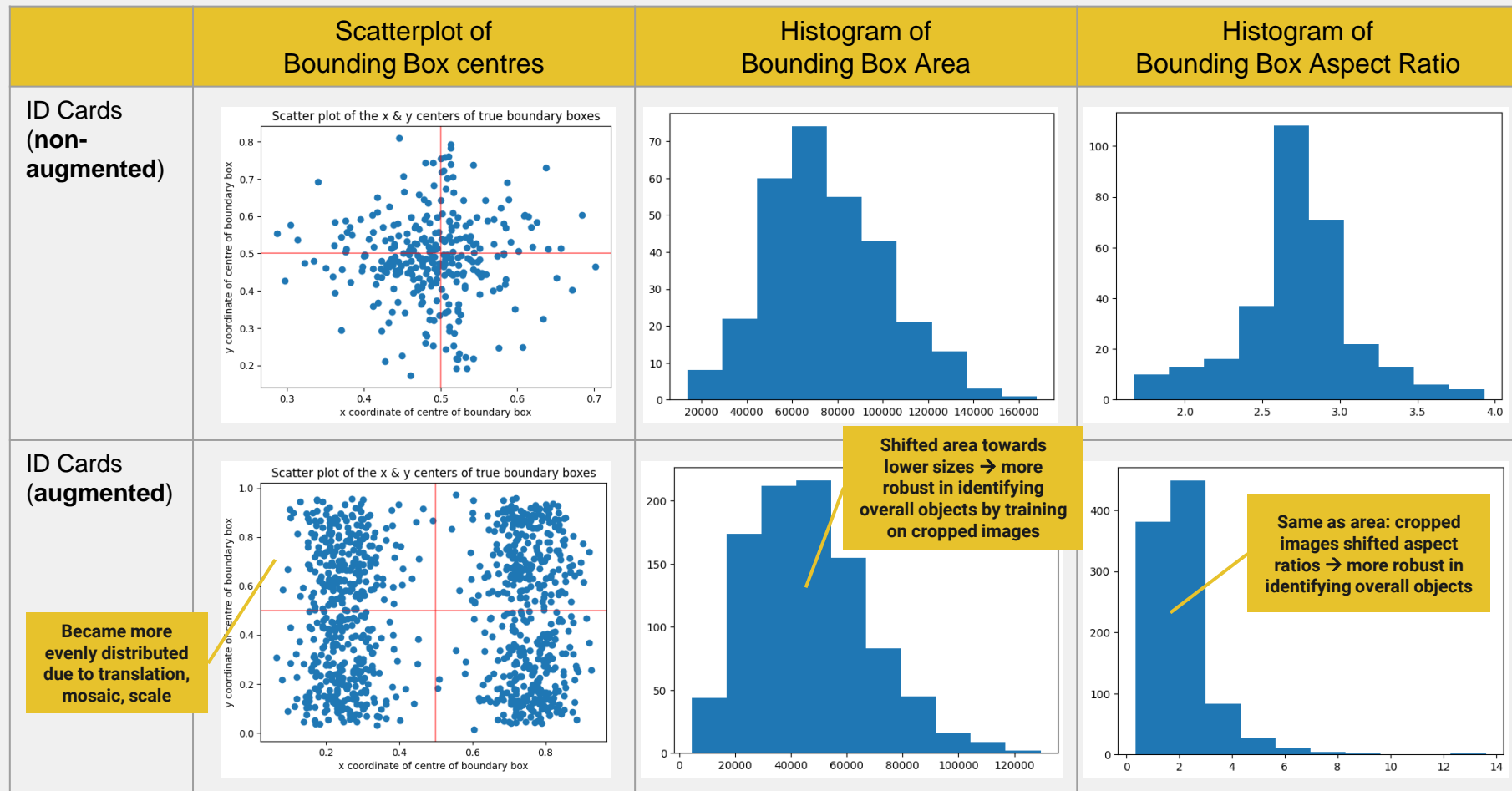
Analysis for Train-Val:

- This model performed much better – likely due to **training data augmentation** (see following EDA).
- Still predicting background as object

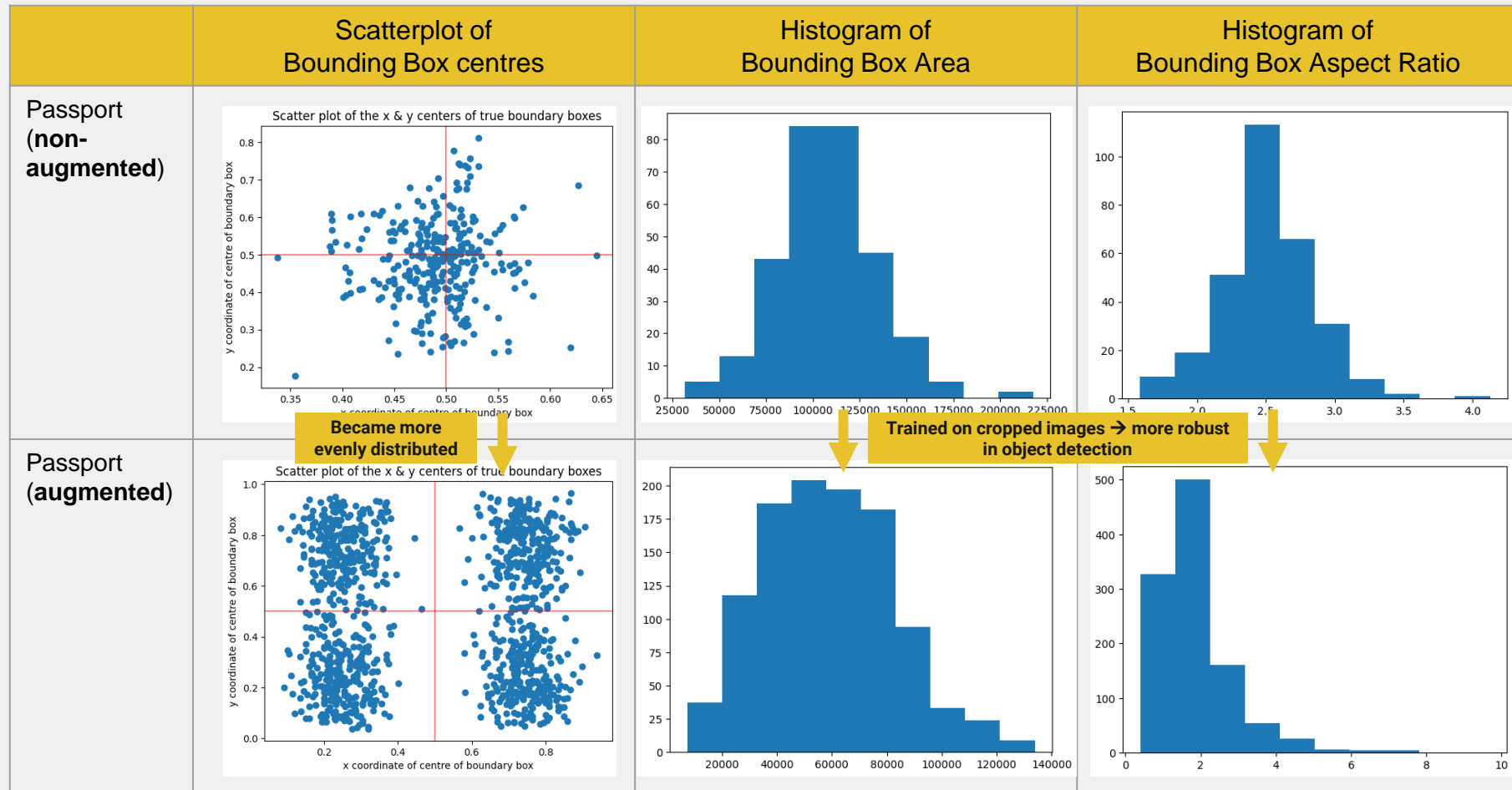
Analysis for Test:

- Better overall performance; performed better for passport prediction

Augmented ID Cards has Better Distributed Samples

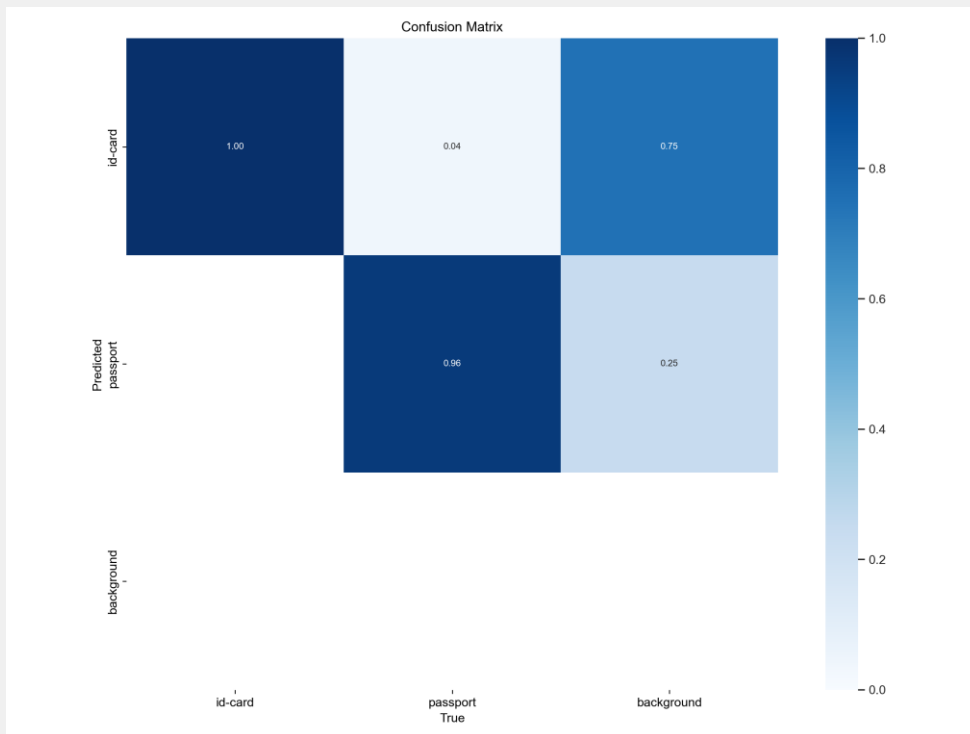


Same Improvements for Passports



Improved Model 2 (+ Augmentation + Rotate + Shear)

Validation Scores



Test Scores

| | ID Cards | Passport |
|---------------------------------------|----------|----------|
| Correctly identified | 95 | 98 |
| Wrongly identified as the other class | 3 | 2 |
| Wrongly identified as background | 2 | 0 |

Analysis for Train-Val:

- This model performed similar to improved model 1
- Still predicting background as object

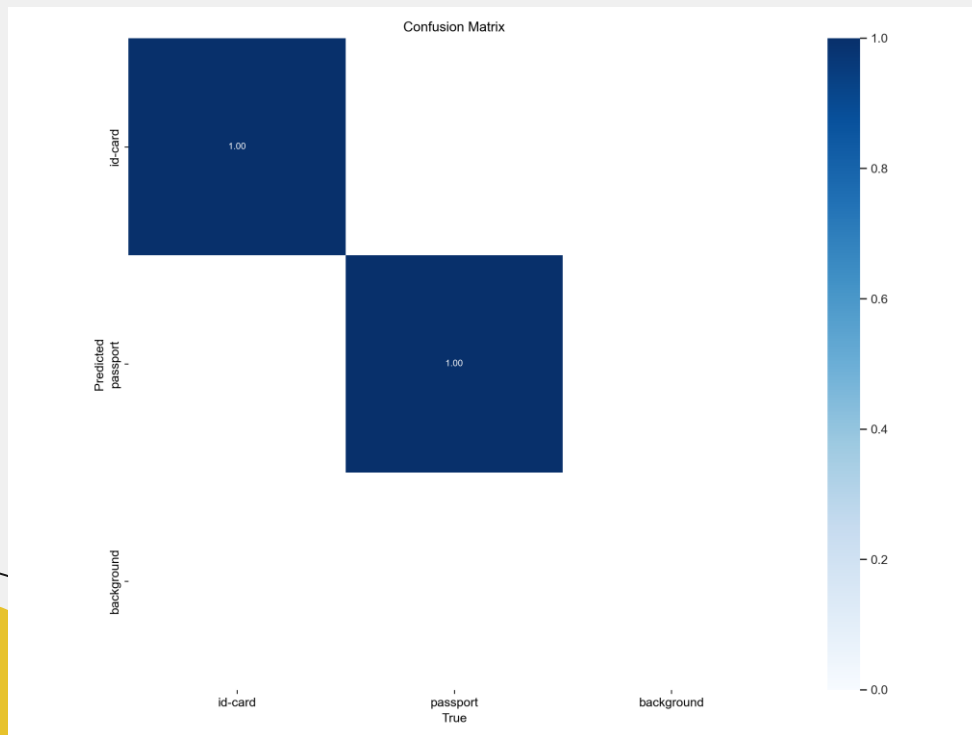
Analysis for Test:

- **Poorer performance than improved model 1** – likely due to excessive data augmentation on small dataset¹¹

[11] Excessive data augmentation lead to poorer model performance, because it introduces data noise ([Ultralytics](#), [Tencent](#))

Improved Model 3 (+ Augmentation + 2x Training Data)

Validation Scores



Test Scores

| | ID Cards | Passport |
|---------------------------------------|----------|----------|
| Correctly identified | 96 | 100 |
| Wrongly identified as the other class | 1 | 0 |
| Wrongly identified as background | 3 | 0 |

Analysis for Train-Val:

- This model performed well

Analysis for Test:

- **Best performance thus far**, although only a small improvement from improved model 1
- **Increasing data size through augmentation can lead to some improvements**¹²

Model Test Performance Comparison

| | Baseline | | Improved 1 | | Improved 2 | | Improved 3 *best performance* | |
|--|----------|----------|------------|----------|------------|----------|----------------------------------|----------|
| | ID Cards | Passport | ID Cards | Passport | ID Cards | Passport | ID Cards | Passport |
| Correctly identified | 99 | 62 | 95 | 99 | 95 | 98 | 96 | 100 |
| Wrongly identified as the other class | 1 | 25 | 5 | 1 | 3 | 2 | 1 | 0 |
| Wrongly identified as background | 0 | 13 | 0 | 0 | 2 | 0 | 3 | 0 |
| Total correct predictions | 161 | | 194 | | 193 | | 196 | |
| Total wrong predictions | 39 | | 6 | | 7 | | 4 | |

Streamlit Demonstration



**Please scan
the QR code to access
my Streamlit app!**



Conclusion



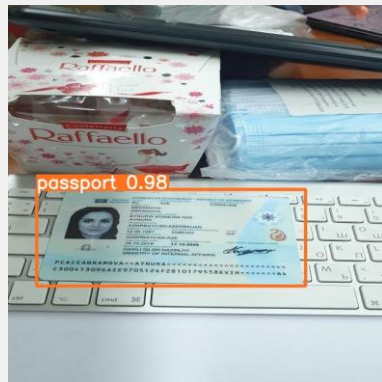
Summary

Problem statement

A digital bank in Singapore wants to expand its banking business into Indonesia, but **faced the challenge of Indonesia not having digital national ID cards**. The bank wants to **avoid manual ID verification for account opening**, because it leads to a poor customer experience

Results

As tasked by the bank, an **object detection model was successfully developed** with a test accuracy of **96% for ID cards** and **100% for passports**. The streamlit deployment also achieved a classification speed of **~3 seconds** for each photo submission¹³.



[13] tested on both PC (using CPU) and mobile phone (Samsung Galaxy S22+)

Limitations & Areas for Improvement

Limitations

- **Dataset used is too small** even for transfer learning, likely leading to **poor generalisation**
- **Possibly limited performance on Southeast Asian ID documents** (due to nature of dataset)
- Current model is **unable to discern quality of submissions** (e.g. blur, too dark, too bright etc.)

Areas for improvement

- **Conduct proof-of-concept using real data** ($\geq 1,500$ images per class)
- **Use transfer learning to train on local dataset** (i.e. start from developed weights)
- **Develop a multi-head¹⁴ model to conduct quality classification** (each head to classify 1 type of quality)

[14] [Cornell, Sep 2021](#), [Marvik, Dec 2022](#)



Thank You!



Do you have any questions?



Annex: Dataset

- The digitally generated photos were printed and made into physical documents
- These documents were then taken under these conditions:
 - Low lighting conditions (20 documents of each type)
 - Keyboard as a background (10 documents of each type)
 - Natural lighting, captured outdoors (10 documents of each type)
 - Table as a background (10 documents of each type)
 - Cloth with various textures as a background (10 documents of each type)
 - Text document as a background (10 documents of each type)
 - High projective distortions of the document (20 documents of each type)
 - Highlight from the sun or lamp hides a portion of the document (10 documents of each type)
- Each country abide to these pre-set parameters:
 - 80% are adults (18 – 60 years old)
 - 10% are seniors (>60)
 - 10% are children, adolescents (≤ 17)
 - 50:50 male-female ratio
- Source: MIDV-2020: A Comprehensive Benchmark Dataset for Identity Document Analysis

Annex: Brief History of YOLO

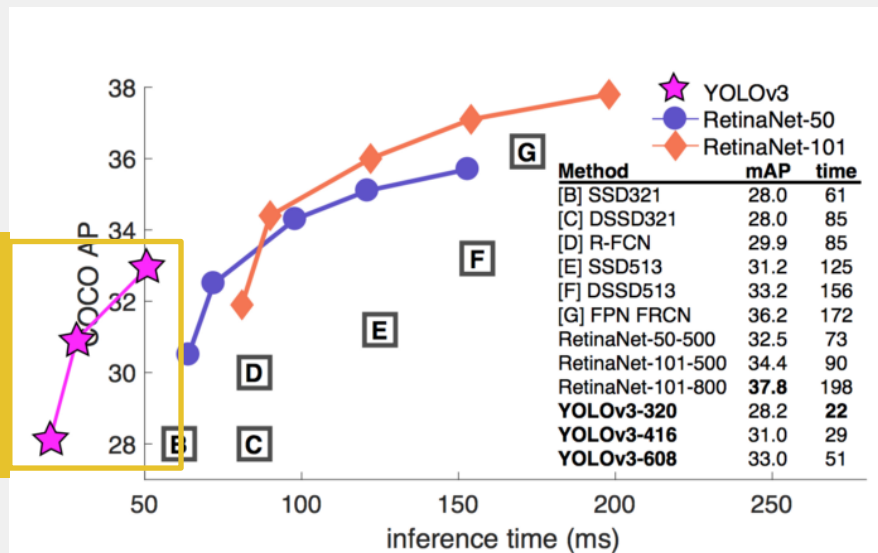
- **YOLO (You Only Look Once)** is a popular object detection and image segmentation model developed by Joseph Redmon and Ali Farhadi at the University of Washington. The first version of YOLO was released in 2015 and **quickly gained popularity due to its high speed and accuracy**.
- YOLOv2 was released in 2016 and improved upon the original model by incorporating batch normalization, anchor boxes, and dimension clusters. YOLOv3 was released in 2018 and further improved the model's performance by using a more efficient backbone network, adding a feature pyramid, and making use of focal loss.
- In 2020, YOLOv4 was released which introduced a number of innovations such as the use of Mosaic data augmentation, a new anchor-free detection head, and a new loss function.
- In 2021, Ultralytics released YOLOv5, which further improved the model's performance and added new features such as support for panoptic segmentation and object tracking. In addition, Ultralytics maintains a public open-source repository to provide guides and tutorials on the installation and use of YOLOv5 – **making YOLOv5 easy to build and use**.

Source: [Ultralytics](#)



Annex: Why YOLO and Not Other Models?

YOLO excels in detection speed, enabling object detection in real-time videos



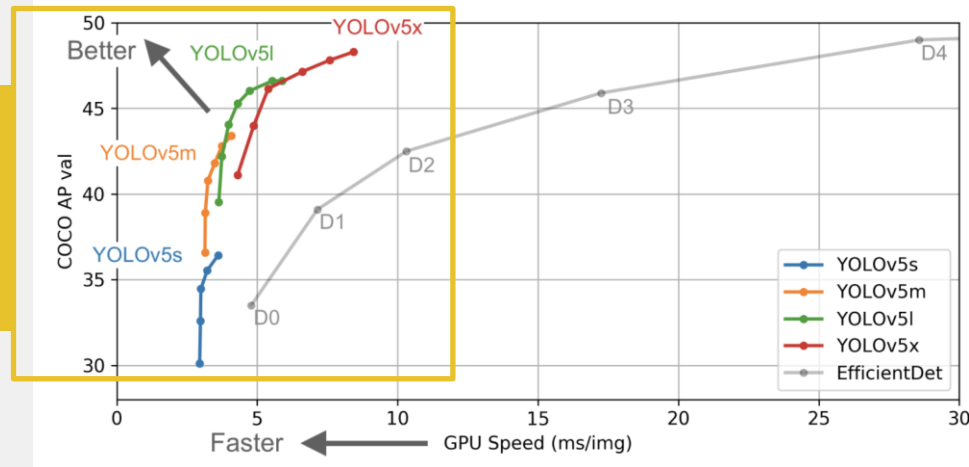
Source: [YOLOv3 paper](#)

YOLO is a **single-shot detector*** that uses a fully convolutional neural network (CNN) to process an image. It processes an entire image in a single pass, making them computationally efficient.

*Single-shot object detection uses a single pass of the input image to make predictions about the presence and location of objects in the image. Whereas a two-shot object detection uses two passes of the input image to make predictions about the presence and location of objects: the first pass is used to generate a set of proposals or potential object locations, and the second pass is used to refine these proposals and make final predictions

Annex: Why YOLOv5l (Large)?

**YOLOv5l offers a
good balance
between
performance and
speed of inference**

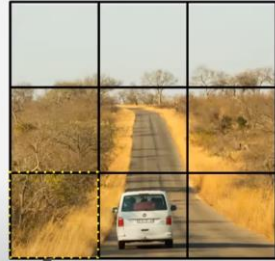


Source: [Ultralytics](#)

Annex: YOLOv5 Overview

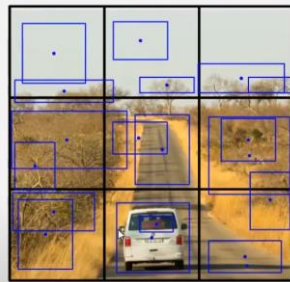
Understanding the training input / prediction output tensor (2)

1. Divide the image into cells with an $S \times S$ grid.
2. Each cell predicts B bounding boxes.
3. Return bounding boxes above confidence threshold.

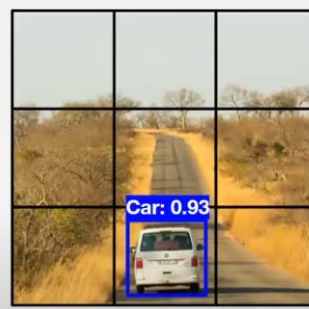


Cell

$S = 3$



$B = 2$

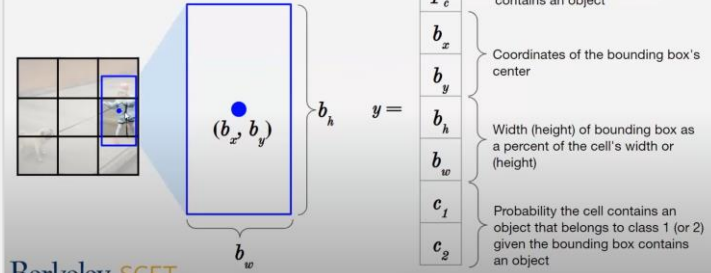


$B = 2$ means each cell is responsible for predicting 2 bounding boxes

In practice, larger values of S & B are used (e.g. $S = 19$ and $B = 5$) to identify more objects in an image

Let's use a simple example where there are 3×3 cells ($S=3$), each cell predicts 1 bounding box ($B=1$), and objects are either dog = 1 or human = 2.

For each cell, the CNN predicts a vector y :



Example:

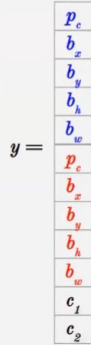
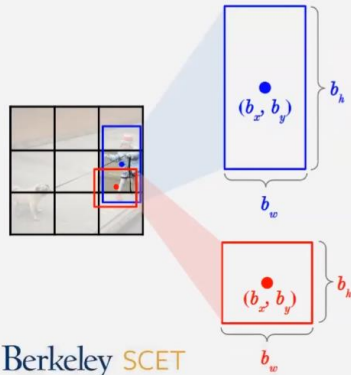
| |
|-------|
| 1 |
| b_x |
| b_y |
| b_h |
| b_w |
| 0 |
| 1 |

Annex: YOLOv5 Overview

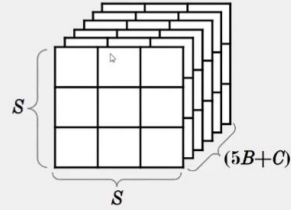
Understanding the training input / prediction output tensor (2)

Encoding Multiple Bounding Boxes

What happens if we predict multiple bounding boxes per cell ($B > 1$)? We simply augment y .

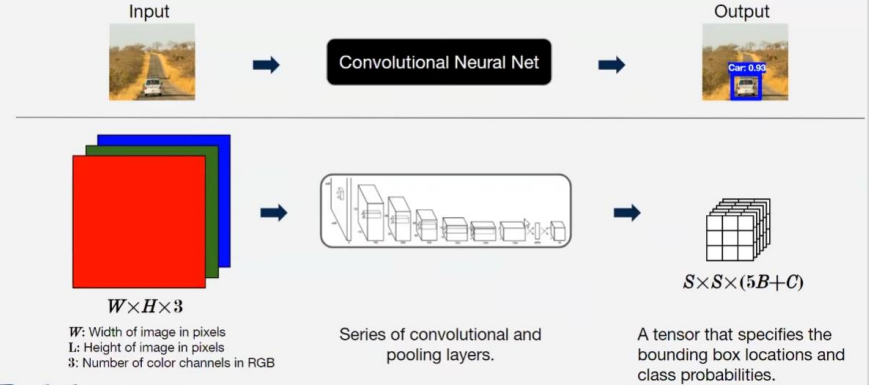


The CNN will predict a y for each cell, so the size of the output tensor (multidimensional "matrix") should be: $S \times S \times (5B+C)$



Notice that y has $5B+C$ elements (C is the number of classes).

YOLO Overview

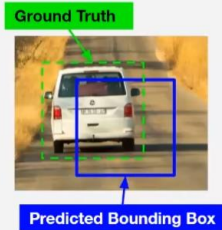


Annex: YOLOv5 Overview

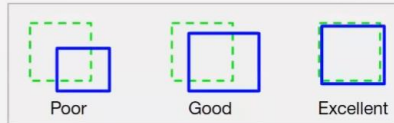
Non-Max Suppression: how to eliminate overlapping bounding boxes

Measuring Performance with UoI

- **Union over Intersection (UoI)** measures the overlap between two bounding boxes.
- During training, we calculate the UoI between a predicted bounding box and the ground truth (the pre-labeled bounding box we aim to match)



$$\text{Union over Intersection} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$



<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

Double Counting Objects (Non-Max Suppression)

- Sometimes the same object will be detected multiple times
- **Non-max suppression** solves multiple counting by removing the box with the lower confidence probability when the UoI between 2 boxes with the same label is above some threshold.



UoI: 0.62

UoI: 0.47

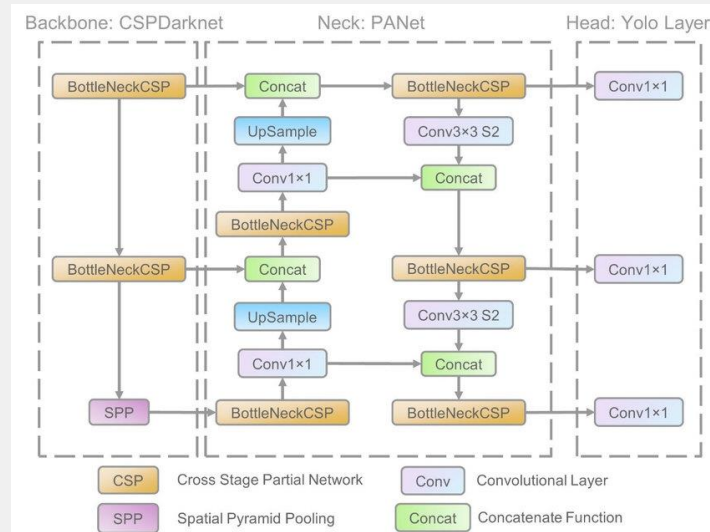


1. Identify the box with the highest confidence.
2. Calculate the UoI between the highest confidence box and each of the other boxes.
3. Suppress boxes with UoI above a selected threshold (usually 0.3)

Annex: YOLOv5 Architecture

The YOLO network consists of three main pieces:

- **Backbone:** A convolutional neural network that aggregates and forms image features at different granularities.
 - YOLOv5 uses a Cross Stage Partial Network (CSPNet) architecture
- **Neck:** A series of layers to mix and combine image features to pass them forward to prediction.
 - YOLOv5 uses PA-NET neck for feature aggregation
- **Head:** Consumes features from the neck and takes box and class prediction steps.



Annex: YOLOv5 Architecture

The YOLO network consists of three main pieces.

- **Backbone:** A convolutional neural network that aggregates and forms image features at different granularities.
- **Neck:** A series of layers to mix and combine image features to pass them forward to prediction.
- **Head:** Consumes features from the neck and takes box and class prediction steps.

<https://towardsai.net/p/computer-vision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-and-demystified>

<https://iq.opengenus.org/yolov5/>

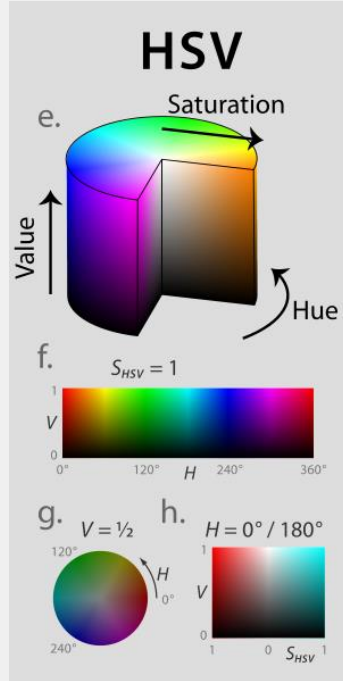
<https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

Annex: YOLOv5 Loss Functions

- Classes loss(BCE loss)
- Objectness loss(BCE loss)
- Location loss(CIoU loss)



Annex: Colour Augmentation



- **Saturation: intensity of colour in an image.** Highly saturated images have vivid, rich colours and lowly saturated images are pale and washed-out
- **Hue: attribute of a visible light** due to which it is differentiated from or similar to the primary colors: red, green and blue
- **Value (a.k.a brightness): the perception of how intense the light coming from a screen is**

Source: Techopedia for [saturation](#), [hue](#), [value](#); [Wikipedia](#)



Annex: Dimension Augmentation (1)

- **Translation:** vertical / horizontal shift of object



From the left, we have the original image, the image translated to the right, and the image translated upwards.

- **Scale:** altering the image size to be larger (i.e. zoom in) or smaller (i.e. zoom out) than the original



From the left, we have the original image, the image scaled outward by 10%, and the image scaled outward by 20%

Source: [Nanonets](#)



Annex: Dimension Augmentation (2)

- **Flip left-right:** flipping the image horizontally



From the left, we have the original image, followed by the image flipped horizontally, and then the image flipped vertically.

- **Mosaic:** combining a corner of an image with 3 separate corners of 3 other images



aug_-319215602_0_-238783579.jpg



aug_-1271888501_0_-749611674.jpg



aug_1462167959_0_-1659206634.jpg



aug_1474493600_0_-45389312.jpg



aug_1715045541_0_603913529.jpg



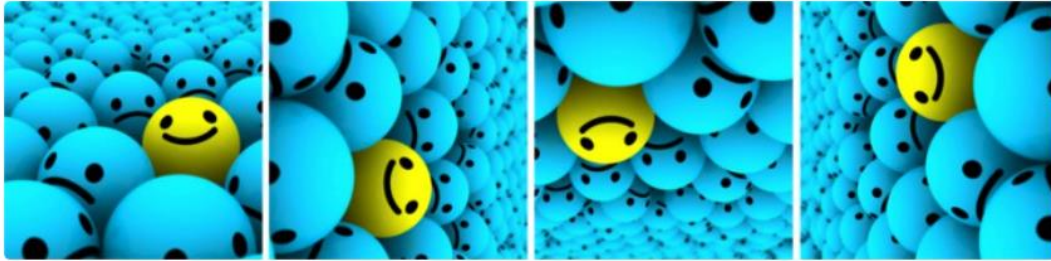
aug_1779424844_0_-589696888.jpg

Source: [Nanonets](#), [Roboflow](#)



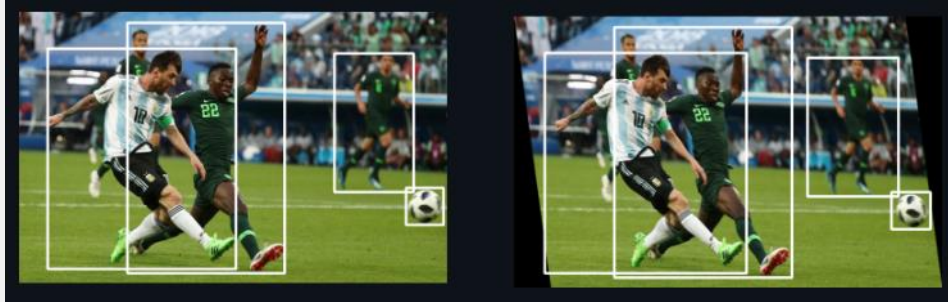
Annex: Dimension Augmentation (3)

- **Rotate:** turn the image around the centre of the image as its rotating axis



The images are rotated by 90 degrees clockwise with respect to the previous one, as we move from left to right.

- **Shear:** displace the image horizontally to different degree, transforming the image into a parallelogram



Source: [Nanonets](#), [Paperspaceblog](#)



Annex: Auto Learning Bounding Box Anchors

- asd



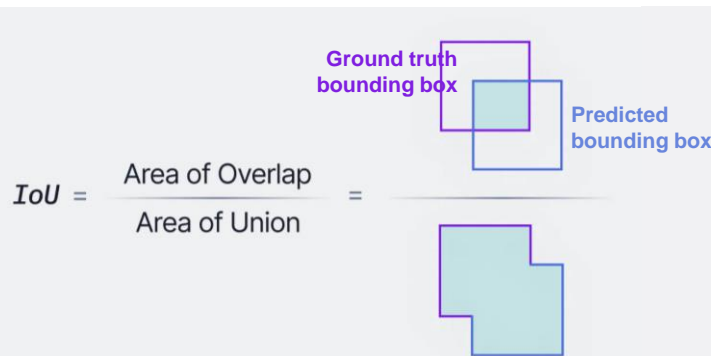
Annex: Intersection over Union (IoU)

Overview

IoU is a popular metric to **measure localization accuracy** and calculate localization errors in object detection models.

Detailed explanation

- The formula for IoU is show on the right
- E.g. IoU score of 0.54 = 54% overlap between the two boxes
- A threshold can be set for the IoU score, to decide how much overlap constitutes a 'positive' classification

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


Source: [v7labs](#)



Annex: Mean Average Precision (mAP)

Overview

mAP is a popular metric that measures the

Detailed explanation

- **Average Precision (AP)** is calculated as the area under a precision vs. recall curve for a set of predictions.
- Recall and precision offer a trade-off that is graphically represented into a curve by varying the classification threshold. The area under this precision vs. recall curve gives us the Average Precision per class for the model. **The average of this value, taken over all classes, is called mean Average Precision (mAP).**

Source: [v7labs](#)

- <https://blog.roboflow.com/mean-average-precision/>
- <https://blog.paperspace.com/mean-average-precision/#:~:text=To%20evaluate%20object%20detection%20models,model%20is%20in%20its%20detections.>

