# Identifying ID Submissions using Computer Vision

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

# Contents

01

# 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 population[1]
- **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[4]:** for account opening, laborious & time-consuming ID verification must be done with digital submission of physical documents (i.e. photos)

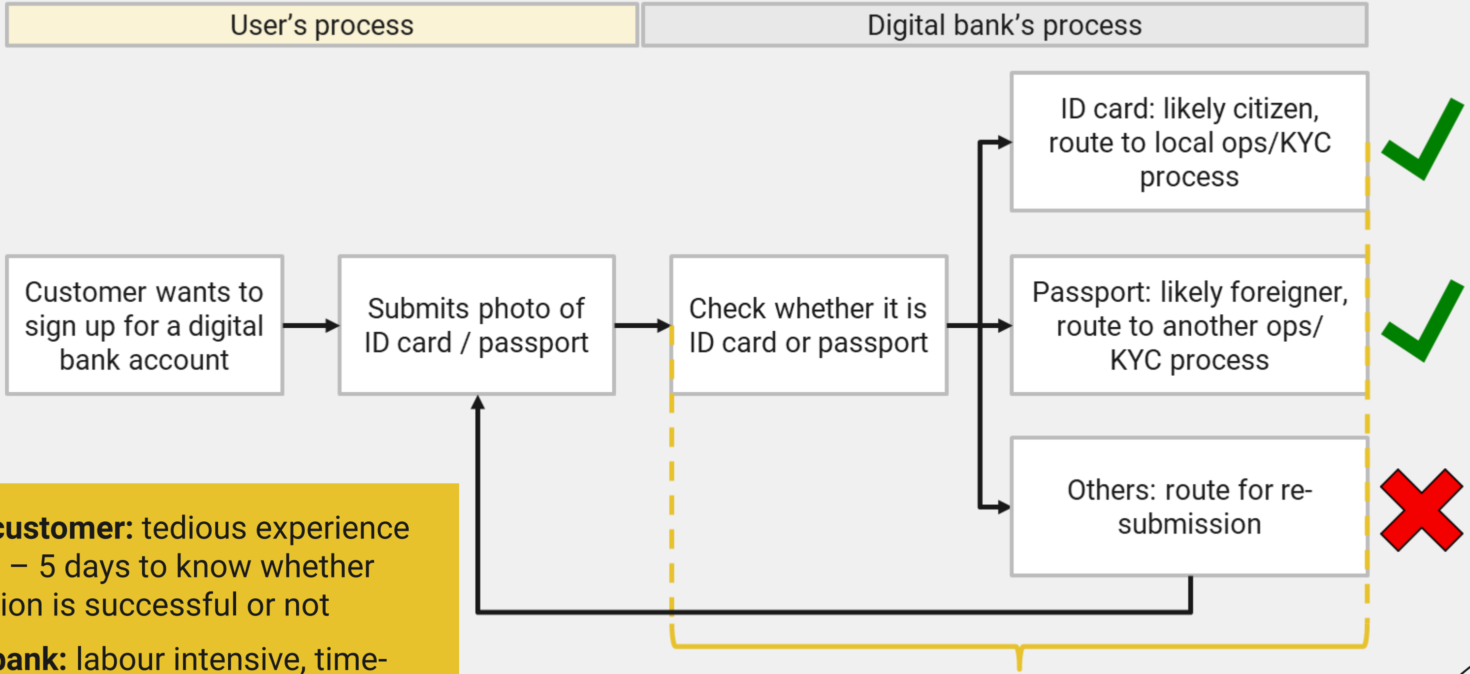
**50%** (~140m) of population yet to have a bank account[1]

**146m** people are between 15 – 49 years of age[3]

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

[1] World Economic Forum, Feb 2022, [2] EY, Dec 2022, [3] United Nations Statistics Division Yearbook, 2020, [4] The World Bank, Nov 2022

# Currently, ID Verification takes: 3 – 5 Working Days

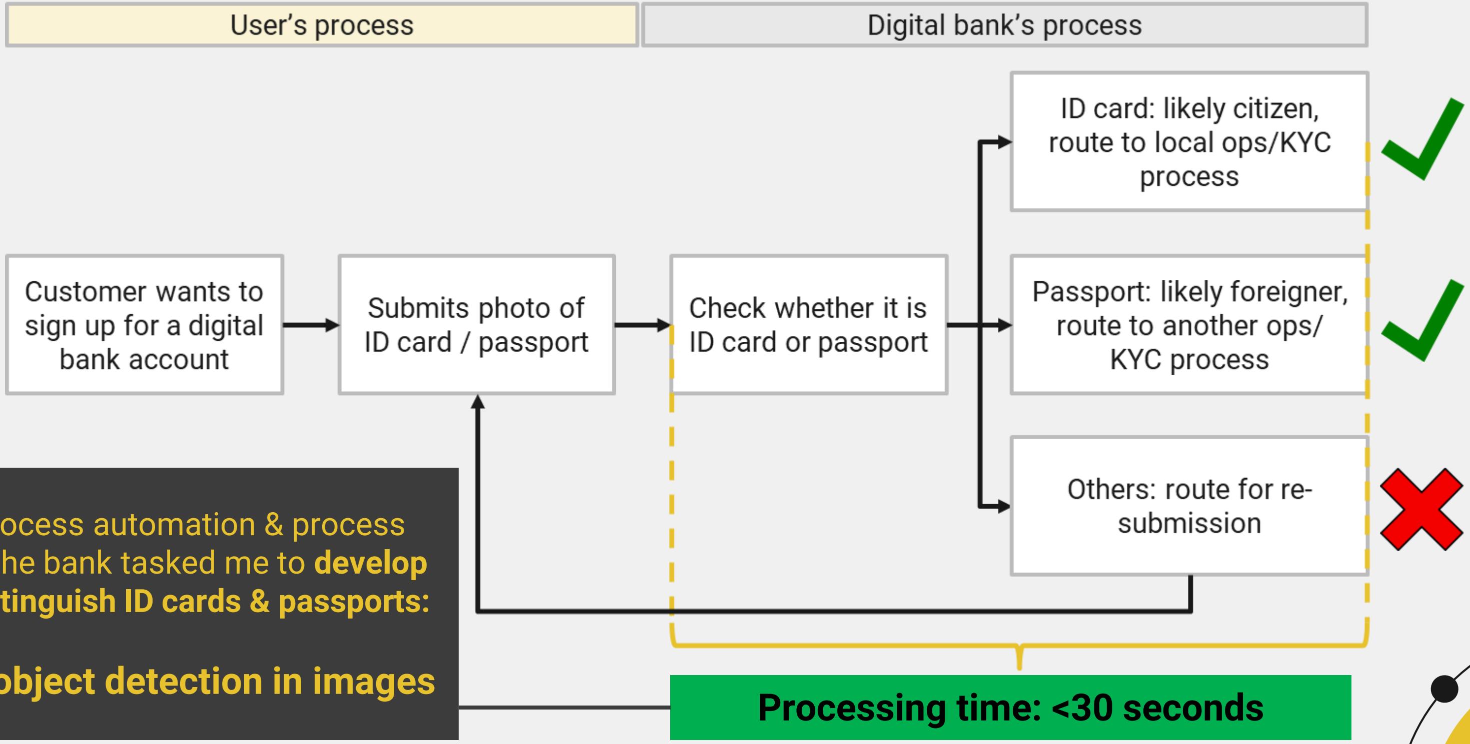



- **For the customer:** tedious experience to wait 3 – 5 days to know whether submission is successful or not
- **For the bank:** labour intensive, time-consuming

**Processing time: 3 – 5 working days[5]**

[5] 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




To enable process automation & process acceleration, the bank tasked me to **develop a model to distinguish ID cards & passports:**

**Multi-class object detection in images**

# 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

02

# Exploratory Data Analysis (EDA)

# Dataset: 1K photos from East Europe & North-West Asia[6]

| | **Train**<br>600 photos | **Validation**<br>200 photos | **Test**<br>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

[6] More details in Annex

# Dataset: Illustration* (1)

**Normal**

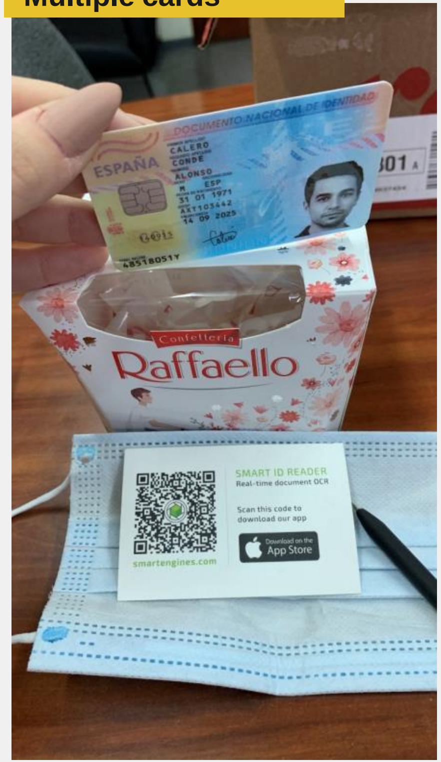
**Different background**

**Different brightness**

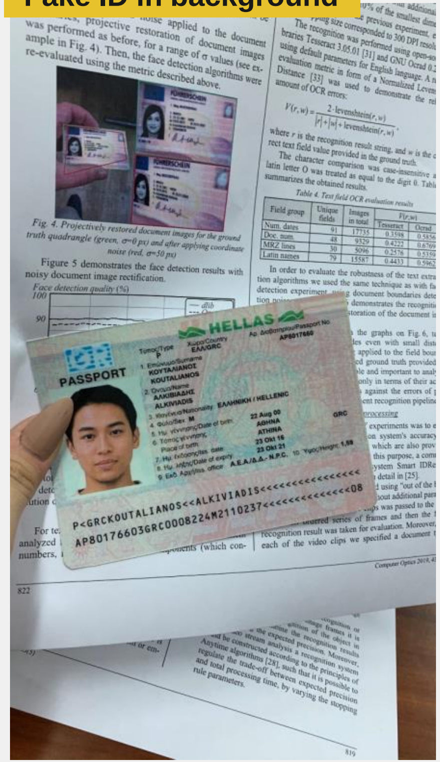
*Illustrated before photo resize to 640 x 640 pixels

# Dataset: Illustration (2)

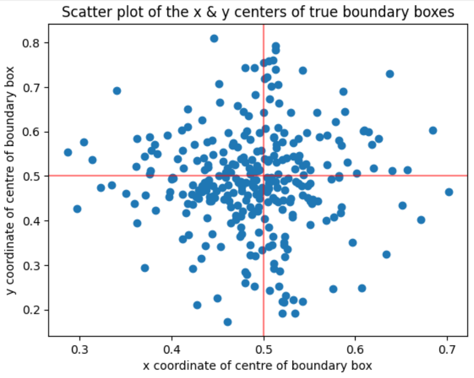
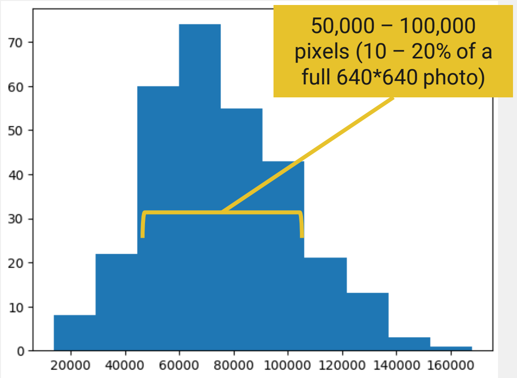
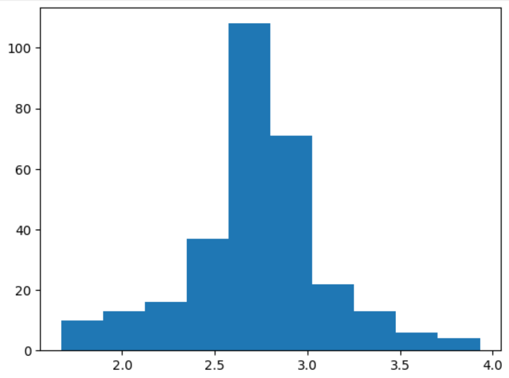
Multiple cards




Fake ID in background



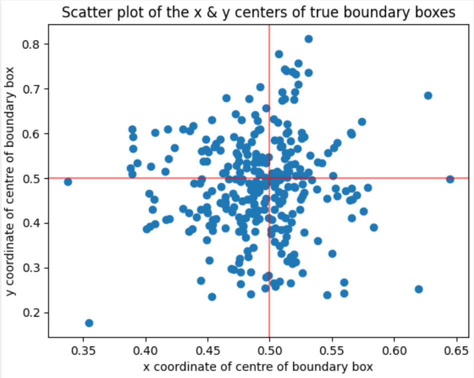
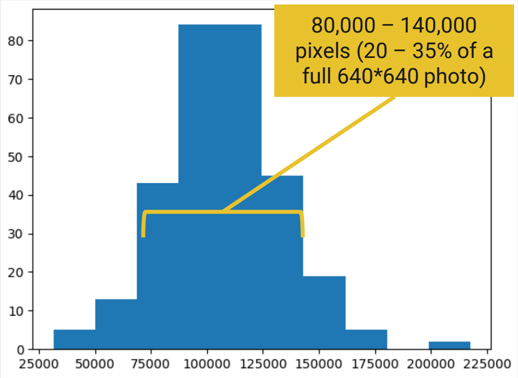
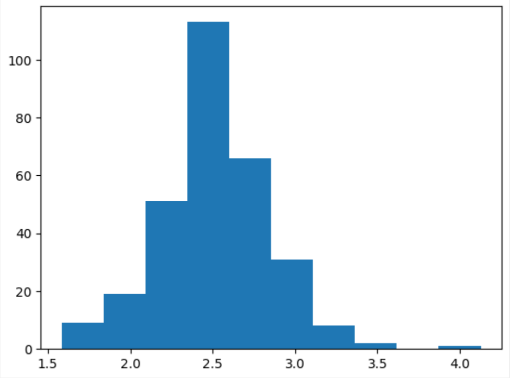
# EDA: Training ID Cards are Usually in the Center, 10-20% of Full Photo Size, have Standard ID Aspect Ratio

<table>
<tr><th></th><th>Scatterplot of<br>Bounding Box centres</th><th>Histogram of<br>Bounding Box Area</th><th>Histogram of<br>Bounding Box Aspect Ratio</th></tr>
<tr><td>Graphs</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td>Analysis</td><td>ID cards usually in centre of photo & along the red lines → trained model may be weak in identifying cards in photo corners</td><td>No samples near 409,600 pixels (640 * 640) → trained model may be weak in identifying close-up photos of ID cards</td><td>Most hover between 2.5 – 3.0 (standard is 2.6 for an ID card*) → trained model may be fixated on the shape of ID cards, instead of details within the cards</td></tr>
</table>

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

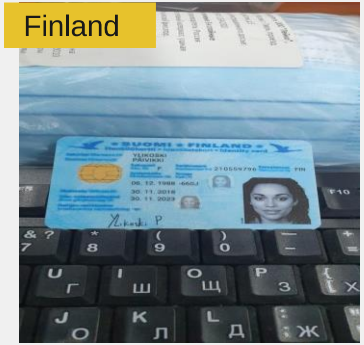
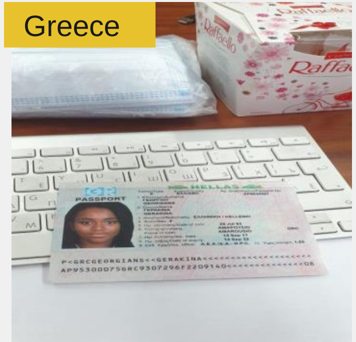
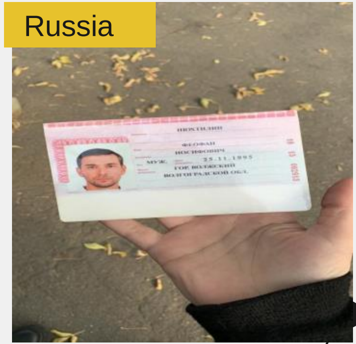
EDA references: Neptune.AI, Kaggle samples

# EDA : Training <u>Passports</u> are Similar to ID Cards

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

*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

03

# Modelling & Deployment

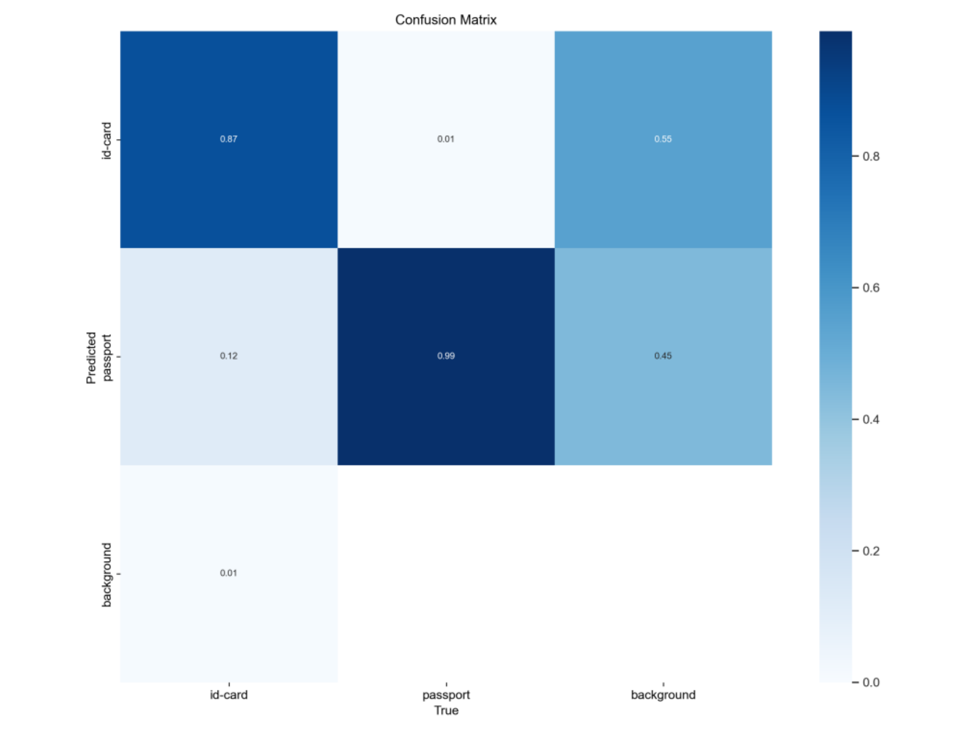
# Modelling Overview

| | 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 | ▪ Colours: hue, saturation, brightness[7]<br>▪ Image translation, scale, flip left-right, mosaic[8] | ▪ Colours: hue, saturation, brightness<br>▪ Image translation, scale, flip left-right, mosaic[8]<br>▪ Rotation, shear[8] | ▪ Colours: hue, saturation, brightness<br>▪ Image translation, scale, flip left-right, mosaic[8] |
| Model training, validation | YOLOv5 Large[9,10,11]:<br>▪ Batch size = 8 (limited by hardware[12])<br>▪ Epochs = 100<br>▪ Starting weights = pretrained weights on COCO 2017 | | | |
| Model testing | ▪ Best weight from training-validation<br>▪ Confidence threshold = 0.7<br>▪ With test-time augmentation | | | |

[7] Detailed explanation of colour augmentation in annex
[8] Detailed explanation of dimensional augmentation in annex
[9] YOLO performance over other models in annex, YOLOv5l performance over other variants in annex
[10] Reference for training parameters for small dataset by Ultralytics, May 2022
[11] No freezing of layers done due to potential performance dip
[12] 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:**

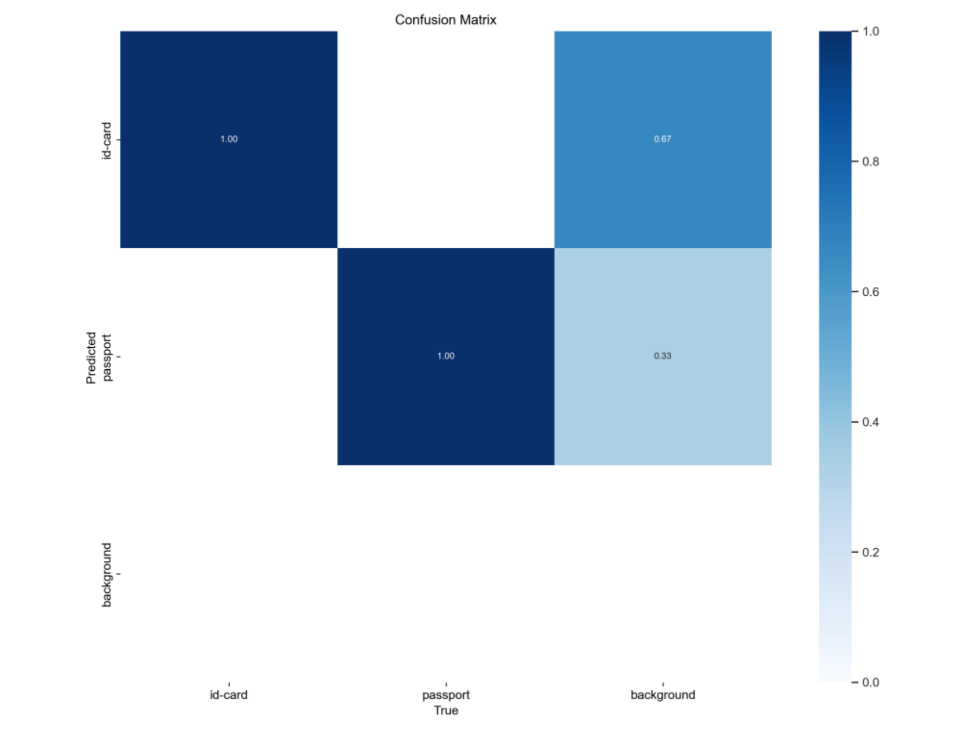
- **Did well for passport, did decently for ID card**
- Identified some background as ID card/Passport

**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 |

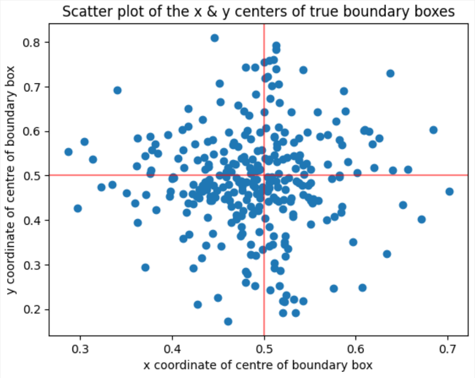
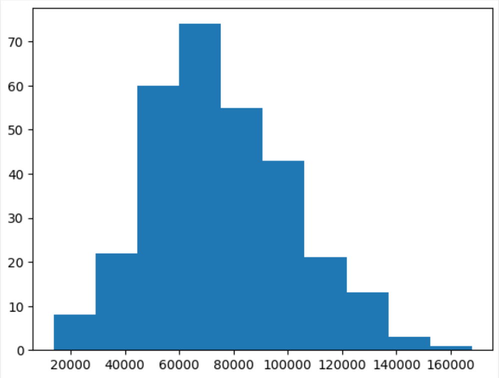
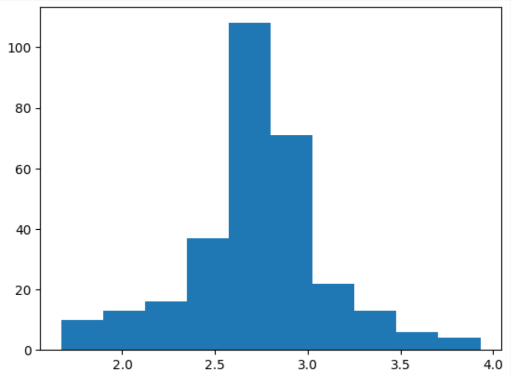
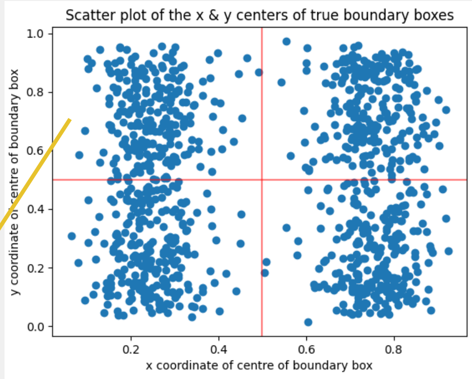
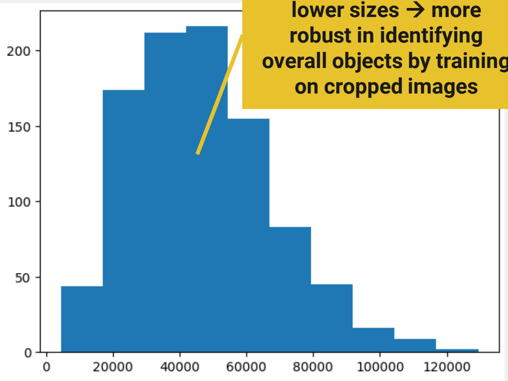
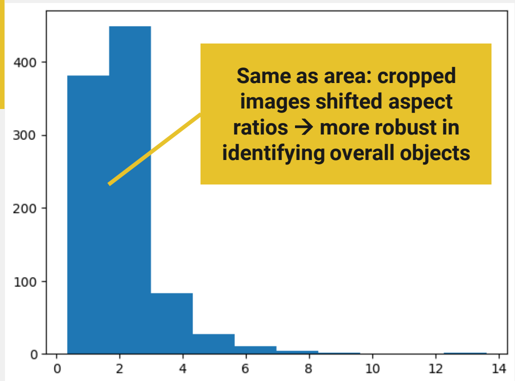
**Analysis for Train-Val:**

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

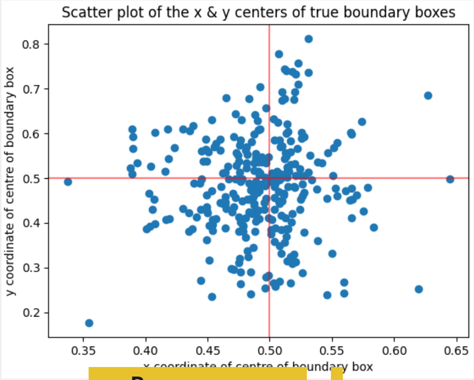
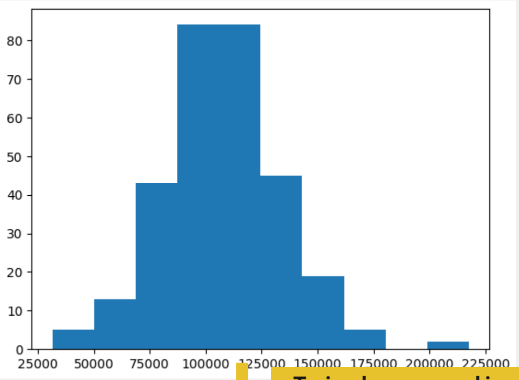
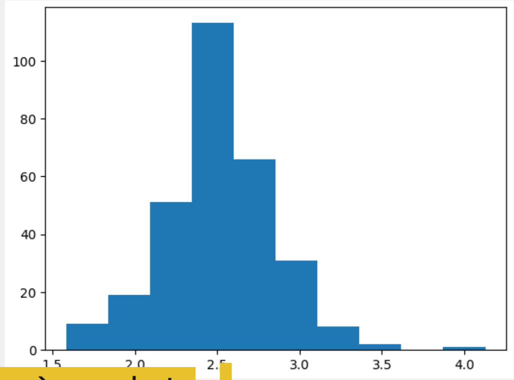
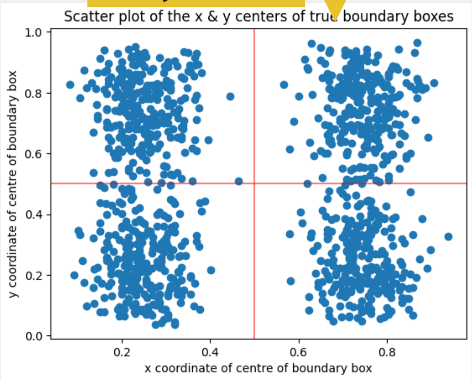
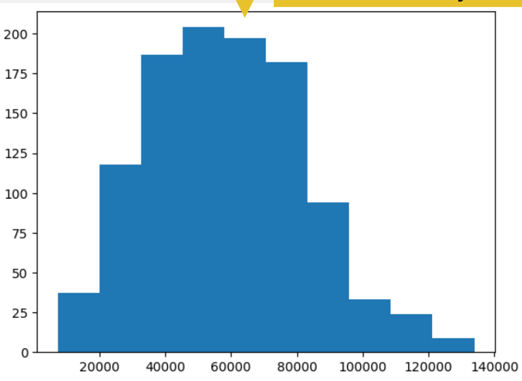
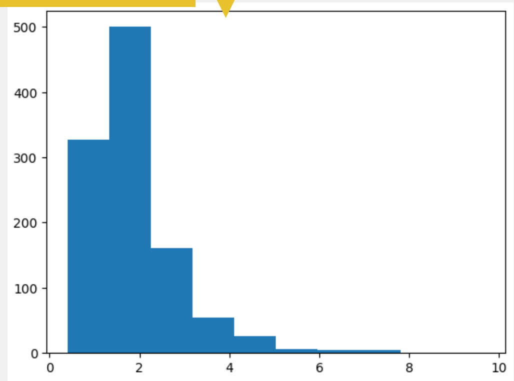
**Analysis for Test:**

- **Better overall performance;** performed **better for passport prediction**

# Augmented ID Cards has Better Distributed Samples

<table>
<tr><th></th><th>Scatterplot of<br>Bounding Box centres</th><th>Histogram of<br>Bounding Box Area</th><th>Histogram of<br>Bounding Box Aspect Ratio</th></tr>
<tr><td>ID Cards<br>(non-augmented)</td><td></td><td></td><td></td></tr>
<tr><td>ID Cards<br>(augmented)<br><br>Became more evenly distributed due to translation, mosaic, scale</td><td></td><td><br>Shifted area towards lower sizes → more robust in identifying overall objects by training on cropped images</td><td><br>Same as area: cropped images shifted aspect ratios → more robust in identifying overall objects</td></tr>
</table>

# Same Improvements for Passports

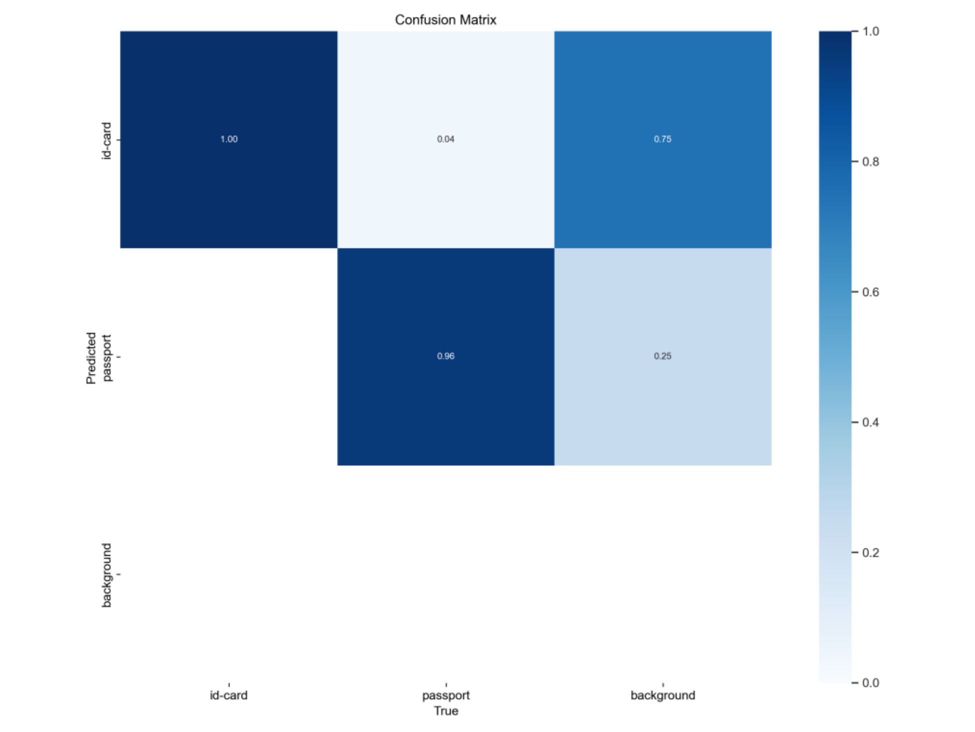
<table>
<tr><th></th><th>Scatterplot of<br>Bounding Box centres</th><th>Histogram of<br>Bounding Box Area</th><th>Histogram of<br>Bounding Box Aspect Ratio</th></tr>
<tr><td>Passport<br>(non-augmented)</td><td>
</td><td></td><td></td></tr>
<tr><td>Passport<br>(augmented)</td><td>
</td><td></td><td></td></tr>
</table>

**Became more evenly distributed**

**Trained on cropped images → more robust in object detection**

# 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

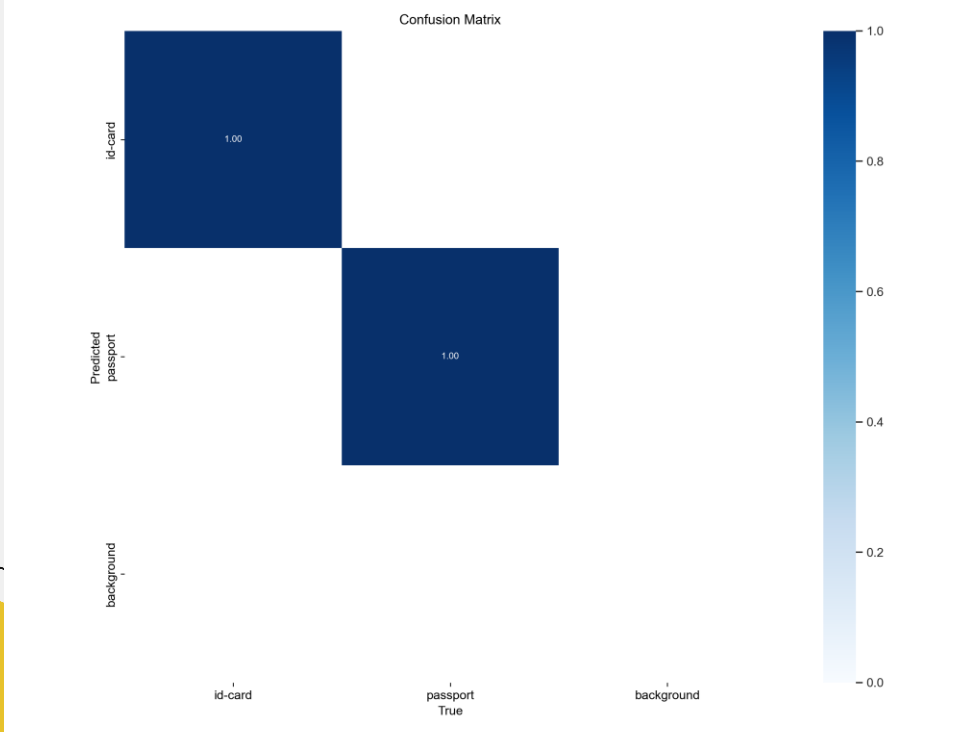
**Analysis for Test:**

- **Poorer performance than improved model 1** – likely due to excessive data augmentation on small dataset[13]

[13] 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**[14]

[14] Small datasets can benefit from data augmentation techniques (The Analytics Club, Datacamp)

# 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!

04

# 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 submission[15]. |





[15] 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** (>= 1,500 images per class)
- **Use transfer learning to train on local dataset** (i.e. start from developed weights)
- **Develop a multi-head**[16] **model to conduct quality classification** (each head to classify 1 type of quality)

[16] 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.

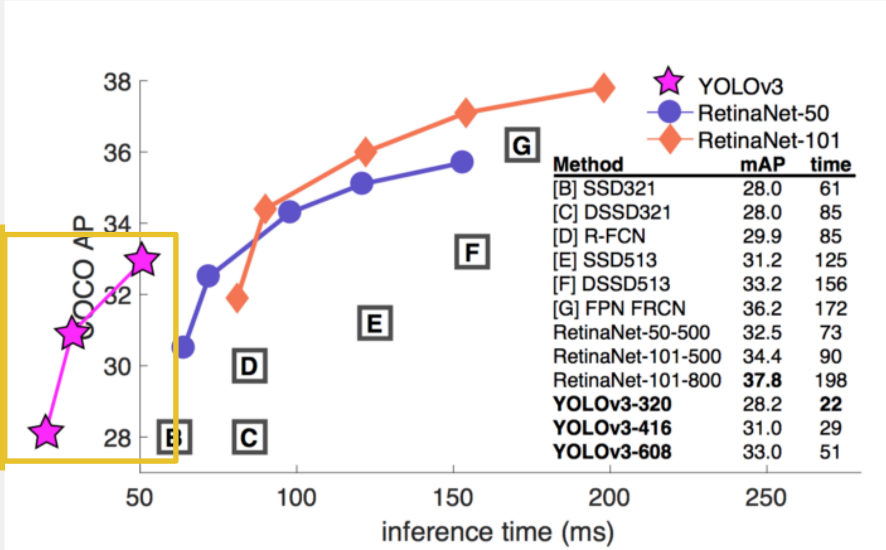
- In 2020, YOLOv4 was released which introduced a number of innovations such as the use of Mosaic data augmentation, a new anchor-free detection head, and a new loss function.

- 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. **YOLOv5 is implemented in Pytorch**, giving more flexibility to control the encoded operations. 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



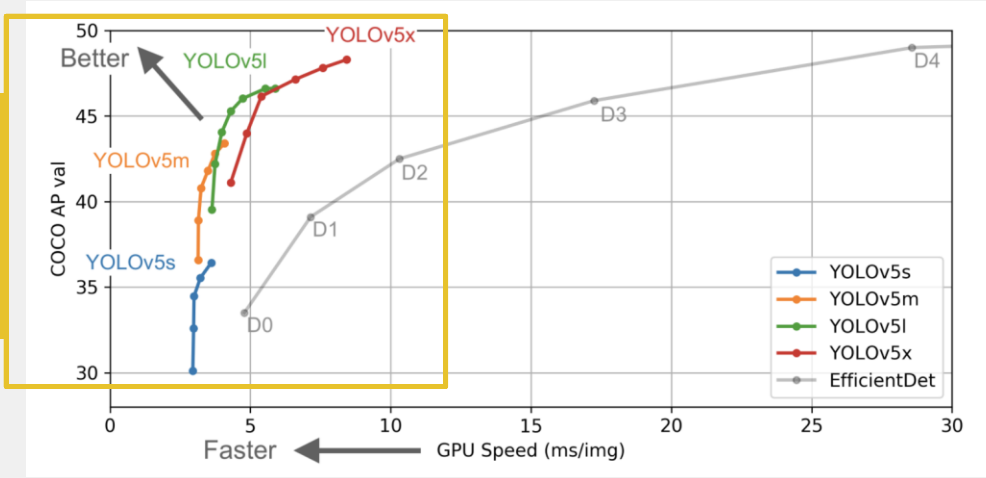

Source: YOLOv3 paper

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)?

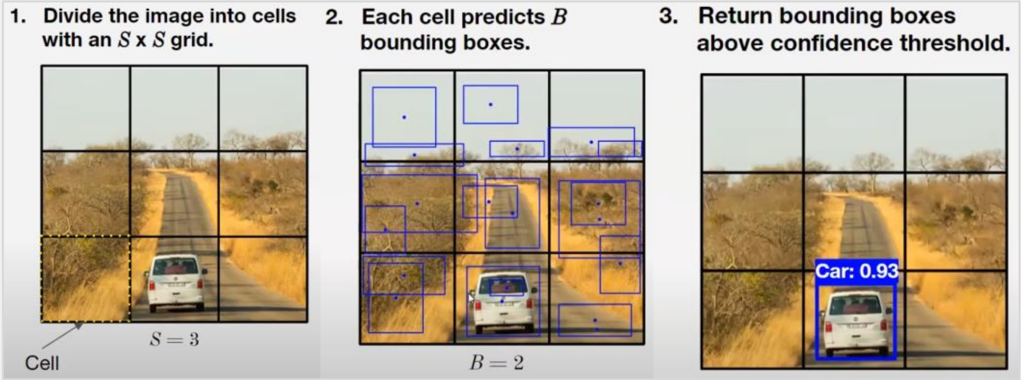
**YOLOv5l offers a good balance between performance and speed** of inference

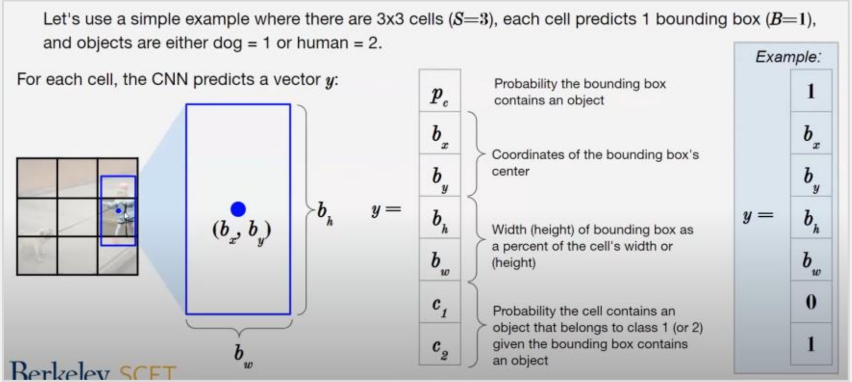



Source: Ultraytics

# Annex: YOLOv5 Overview

## Understanding the prediction output tensor (1)



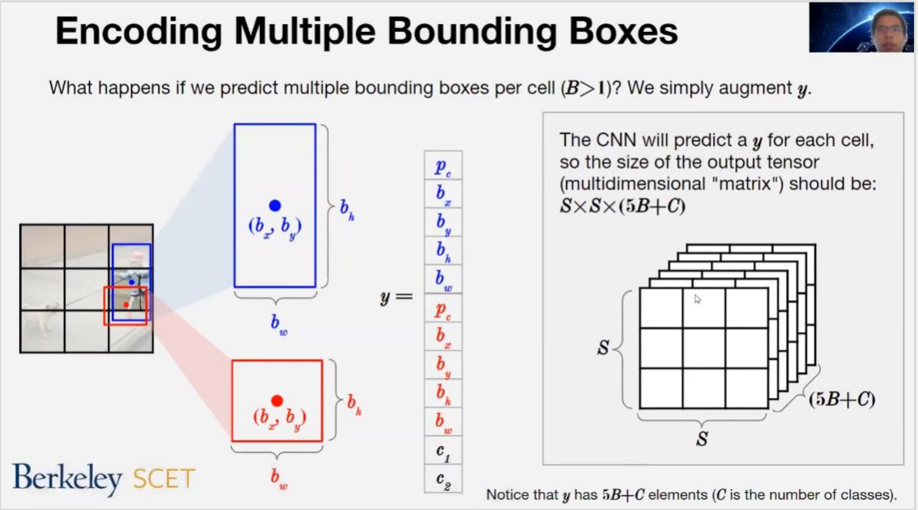

B = 2 means each cell is responsible for predicting 2 bounding boxes

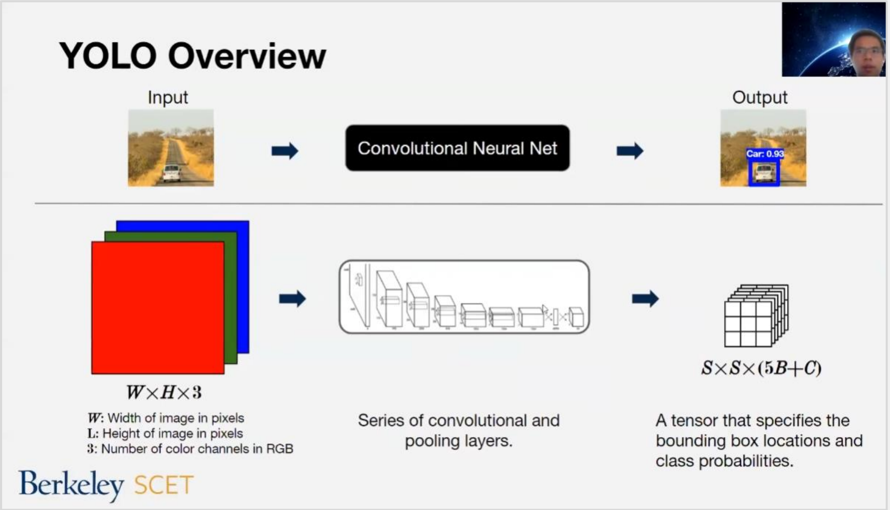
In practice, larger values of S & B are used to identify more objects in an image. YOLOv5l uses 3 multi-scale outputs at strides 8, 16, 32. For a 640 x 640 image, S = 80, 40, 20.

Source: Data-X Berkeley, Aug 2020, Ultralytics

# Annex: YOLOv5 Overview

## Understanding the prediction output tensor (2)



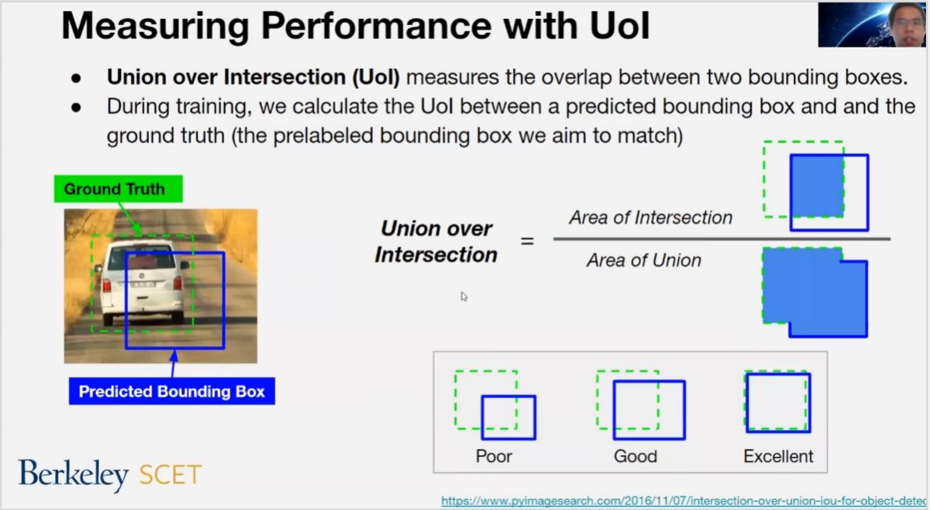

The model output will indicate the predicted class, and the class-specific confidence score:

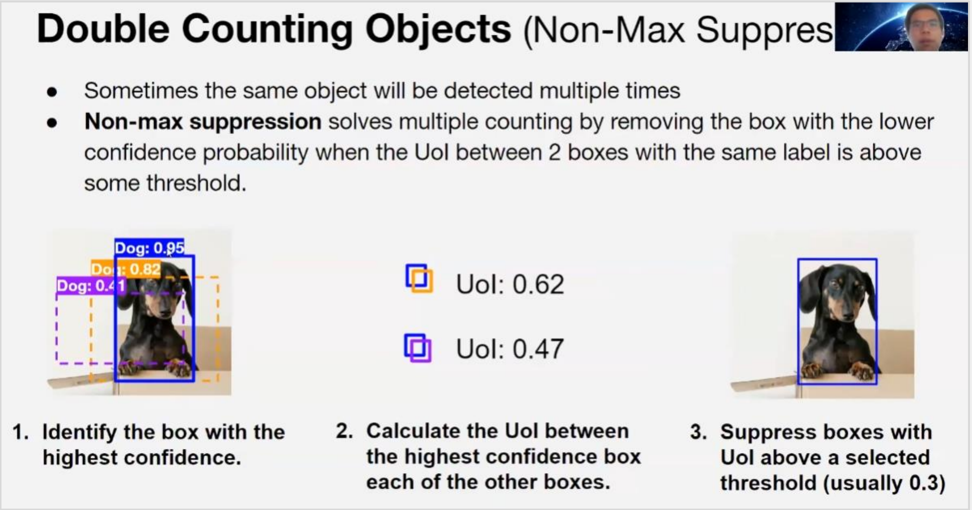
$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}}$$

Source: [Data-X Berkeley, Aug 2020](), [YOLOv1]()

# Annex: YOLOv5 Overview

## Non-Max Suppression: how to eliminate overlapping bounding boxes



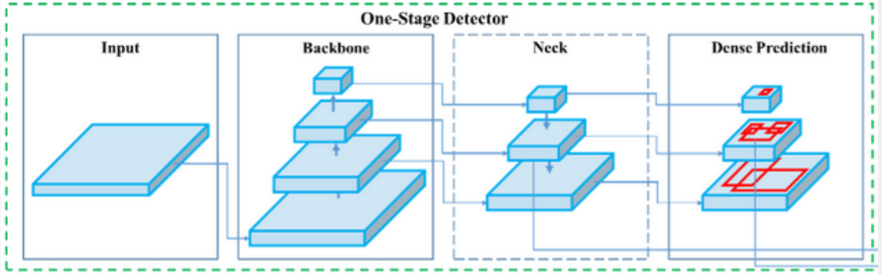




YOLOv5 uses UoL threshold of 0.45

Source: Data-X Berkeley, Aug 2020

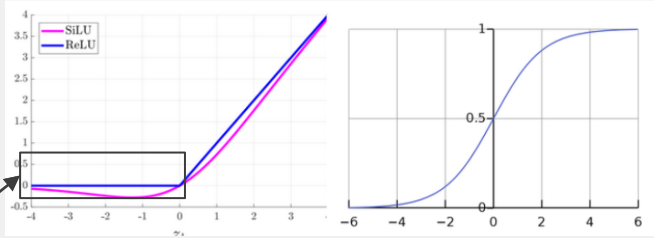
# Annex: YOLOv5 Architecture

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 (mainly
- **Head**: Consumes features from the neck and performs the final stage operations. It applies anchor boxes on feature maps and render the final output: **classes, objectness scores and bounding boxes**.

- **SiLU** (Sigmoid Linear Unit; aka swish) activation function is used with the convolution operations in **the hidden layers**, while the **Sigmoid** activation function is used with the convolution operations in **the output layer**.




To **counter the dying ReLU problem for deep networks**: when most of these neurons return output zero, the gradients fail to flow during backpropagation, and the **weights are not updated**. Ultimately a large part of the network becomes inactive, and it is unable to learn further.

Source: OpenGenus, Towards Data Science

# Annex: YOLOv5 Loss Functions

- YOLOv5 returns three outputs: the classes of the detected objects, their bounding boxes and the objectness scores. Thus, it uses **BCE (Binary Cross Entropy) to compute the classes loss and the objectness loss**. While **CIoU (Complete Intersection over Union) loss to compute the location loss**.

$$Loss = \boxed{\lambda_1 L_{cls} + \lambda_2 L_{obj}} + \boxed{\lambda_3 L_{loc}}$$

## BCE

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- If probability of prediction associated with True class = 1.0, loss = 0
- If probability of prediction associated with True class = ~0, loss = very large value

## CIoU

$$\mathcal{L} = S(\mathcal{B}, \mathcal{B}^{gt}) + D(\mathcal{B}, \mathcal{B}^{gt}) + V(\mathcal{B}, \mathcal{B}^{gt})$$

CIoU loss bounding box regression uses three geometric factors:

- S – the **overlap area** between the predicted box and the ground truth bounding box
- D – the normalised **distance between the central points** of the predicted box and the ground truth bounding box
- V – the difference in **aspect ratios** between the predicted box and the ground truth bounding box

Source: OpenGenus, Towards Data Science, Medium

# Annex: Bounding Box Anchors

State of the art models generally use bounding boxes in the following order:

1. Form thousands of candidate anchor boxes around the image
2. For each anchor box predict some offset from that box as a candidate box
3. Calculate a loss function based on the ground truth example
4. Calculate a probability that a given offset box overlaps with a real object
5. If that probability is greater than 0.5, factor the prediction into the loss function
6. By rewarding and penalizing predicted boxes slowly pull the model towards only localizing true objects

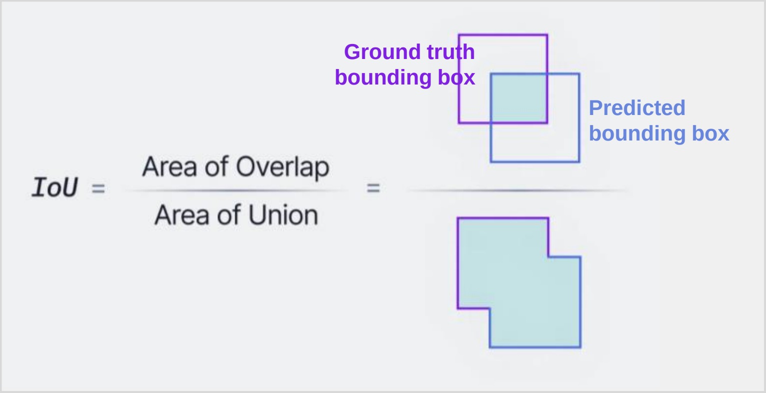
Instead of choosing priors by hand, the **model runs k-means clustering on the training set bounding boxes to automatically find good priors**.

Source: Roboflow, YOLOv2

# Annex: Intersection over Union (IoU)

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

- 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

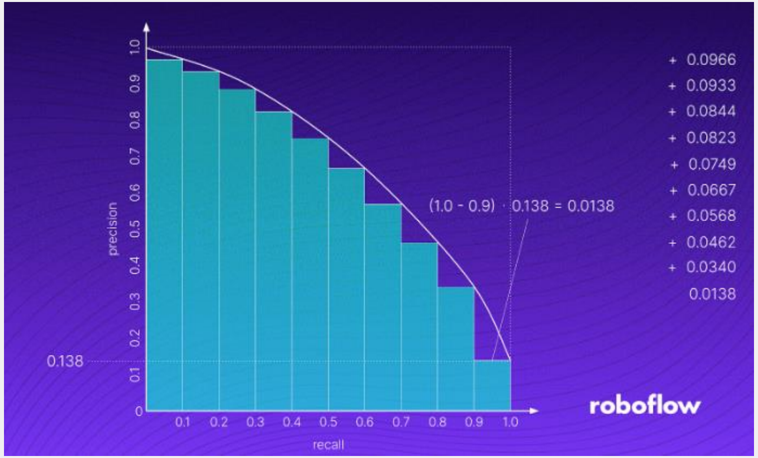



Source: v7labs

# Annex: Mean Average Precision (mAP)

A common evaluation metric used in many object recognition and detection tasks is "mAP", short for "mean average precision". It is a number from 0 to 100; higher value is better.

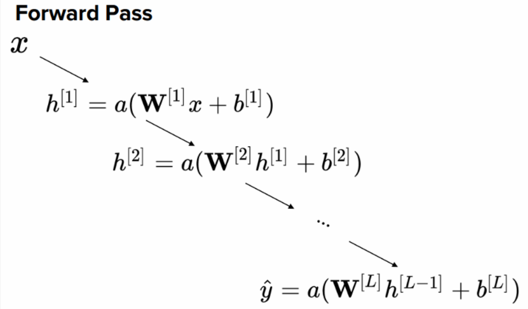
- Combine all detections from all test images to draw a **precision-recall curve (PR curve) for each class**; The "average precision" (AP) is the area under the PR curve.
- Given that target objects are in different classes, we **first compute AP separately for each class, and then average over classes.**
- A detection is a true positive if it has "intersection over union" (IoU) with a ground-truth box greater than some threshold (usually 0.5; if so, the metric is "mAP@0.5")

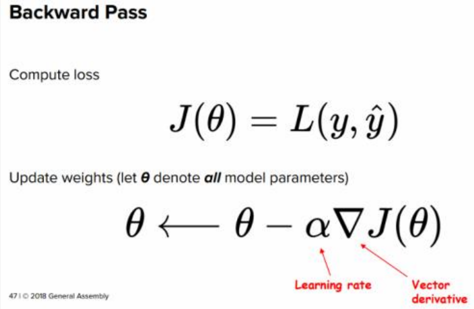



Source: Lil'log

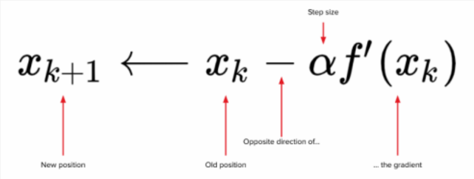
# Annex: Convolutional Neural Network

- **Back propagation**:



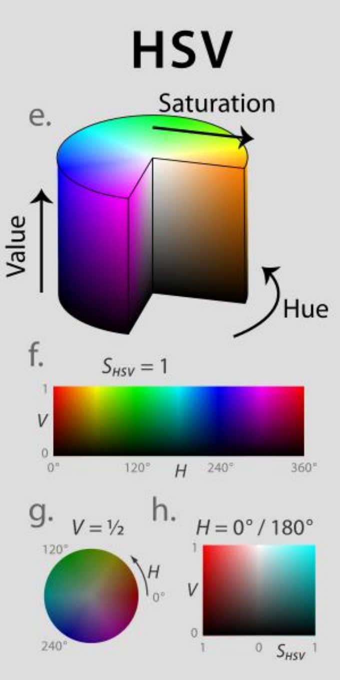




- Optimise weights using **gradient descent**:




- **Loss functions** are measurements of how well the model predicted the outcome
  - Regression: e.g. MSE
  - Binary classification: binary cross-entropy
  - Multiclass classification: cross-entropy
- **Activation functions** are transformation functions applied to the output of a layer
  - E.g. Sigmoid, ReLU

# 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.

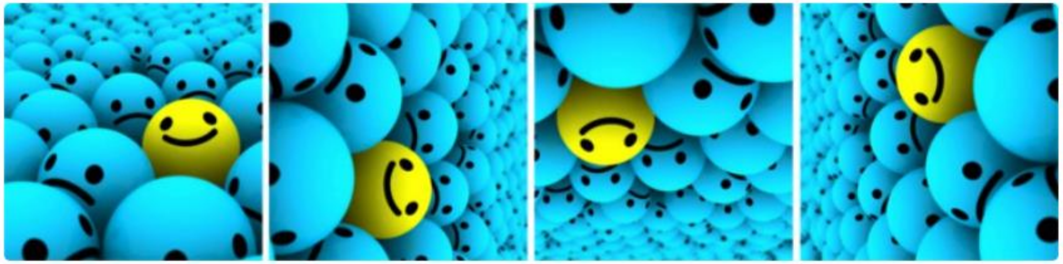
- **Mosaic**: combining a corner of an image with 3 separate corners of 3 other images


[Return to slide 19]

Source: Nanonets

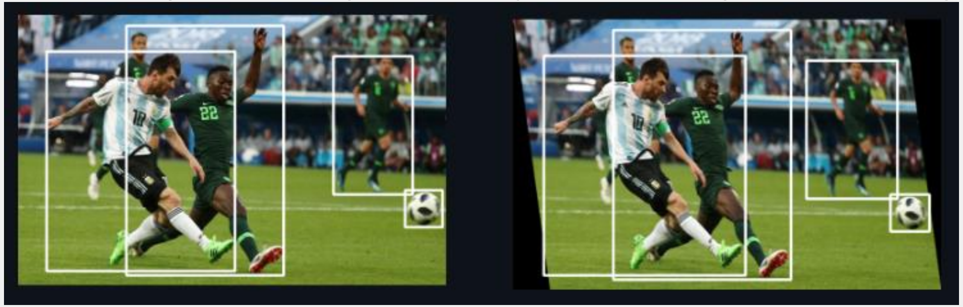
# Annex: Dimension Augmentation (3)

- **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