Customer Analytics: Identification and Classification using YOLO

Senthil Selvaraj / sen07 **Yan Liu / liuyan84** **Chu-Chi Liu / Taiwan**

Abstract

The advent of latest deep learning computer vision techniques has spurred a lot of progress across industries; with great benefits comes risks. Hence the need for a strong customer analytics in crowded locations such as banks, airports and malls. Our paper is focused on YOLO methodology to generate use-cases to identify and classify people. We tested YOLO v2 and the state-of-the art YOLO v3 algorithms to identify object (people) and detect their faces. With dataset that we prepared, we are able to better the original paper's accuracy and achieve 58.78% in detecting people. We adopted CNN model to classify people and achieve a higher accuracy (95.27% for age and 75.74% for gender).

1. Introduction

Almost all public places such as banks and malls have existing infrastructure to capture real time/recorded videos and pictures with closed circuit cameras (CCTV). We plan to leverage this to identify and classify people. In this report, we focus on a use case for a client, let's say managers in a large US bank, to have a better understanding of their customer base with details on number of customers entering and leaving their facilities at any point in time. This will help them not only with operational efficiency to better staff during peak hours but also improve customer satisfaction and suspicious activity monitoring.

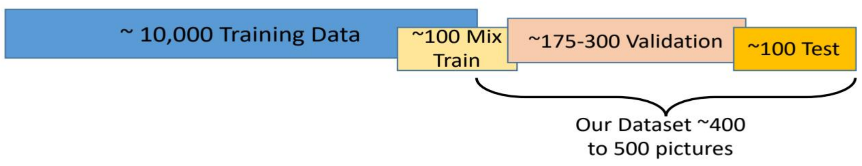
To achieve the above business objective, our project is split into 3 models a) person identification b) face detection and c) classification. People (and other classes such as pets) identification in real time or with still pictures is possible with the improvements in YOLO (You Only Look Once) [1] algorithms and we tested couple of variants of this model – YOLO9000/v2 and YOLOv3 [2]. For classifying the detected people into their gender and age, we used CNN model and its variants.

2. Related Work

Redmon et al.'s YOLO [1][2] real-time object detection network achieves high performance on the PASCAL Visual Object Classes (VOC) 2007 object detection challenge. Their work is inspired by the GoogLeNet for image classification. YOLO is simple: A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. This model solves in an end-to-end fashion two separate problems in computer vision literature: object detection and object classification. YOLO avoids computationally expensive region proposal steps that detectors like Fast R-CNN and Faster-RCNN require [22].

We used Hassner et. al.'s [4] model for gender/age classification. The authors have shown learning representations through the use of deep-convolutional neural networks (CNN), a significant increase in performance can be obtained on these tasks. The model proposed a simple convolutional net architecture that can be used even when the amount of learning data is limited.

3. Dataset and Features



We prepared our dataset, with pictures taken in various public places (i.e. banks) to align closely to our project environment. We hand-labelled the pictures with labelIMG [6] software and used 2 classes (person and pets). Our image dimensions are around 300x300 with resolution 72x72. Input files are in PNG format (though we tried JPEG format as well but resorted to one format) and output annotated files are in Pascal

VOC format as XML files. We modified half of the pictures with cropping and enhancement techniques to bring to a standard format. We have around 400-500 pictures and the three models tweaked the annotated files to suit the individual models as explained below.

Model#1 - YOLO9000/v2 for Identification: We used PASCAL VOC dataset for Darkflow YOLO 9000 model [7]. There is a total of 20 classes with 10,000 total training and test images (typical dimensions of 500x500 in color with 100-200 KB size) with annotated files in XML format. For our model, we used 5100 pictures from this dataset for one iteration and then added 100 of our pictures to align with our environment. We used 300 of 500 of our pictures for Validation/Dev and 100 for Test.

Model#2 - Face Detection Dataset: We used the WIDERFace [8] dataset for YOLOv3. This dataset has a total of 32,203 images containing 393,703 faces with a high degree of variability in scale, pose and occlusion etc. Details of how the dataset fit the models are discussed in methods and results sections in this paper.

Model#3 - Age and Gender Dataset: We used Adience database [5] which has 8 age categories (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-) as shown in table below with label 0 to 7 representing each age group. For gender, the labels are 0-male and 1-female. Since the 500 pictures taken in various public places (i.e. banks) might contain multiple people in a picture with faces not facing the camera, we used opencv and dlib to detect the faces in each picture and manually labelled each of them. Finally, we were able to get about 169 faces for test, which is just about 1% of size of the training set. Moreover, this set is too small, so to avoid overfitting, this is used only in Test set.

Age	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	count
Training set	1387	1511	1723	1356	3049	1573	547	677	11823
Val set	172	157	196	144	320	172	64	59	1284
Test set	1	4	3	21	74	31	28	7	169

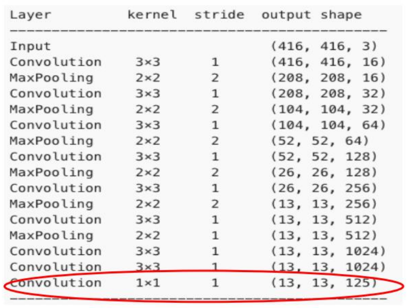
Gender	Male	Female	count
Training set	5568	6688	12256
Validation set	574	765	1339
Test set	57	112	169

4. Methods

Model#1 – Person Identification with YOLO9000 / v2 on PASCAL Dataset:

YOLO9000, a state-of-the-art, real-time object detection system can detect over 9000 object categories. The improved model, YOLOv2, is state-of-the-art on standard detection tasks on PASCAL VOC and COCO and can run at varying sizes, offering an easy tradeoff between speed and accuracy. The accuracy this model achieved was around 57%.

The YOLOV2 Architecture: The network has 24 convolutional layers followed by 2 fully connected layers. The model that we used for our project is Tiny YOLO, which uses 9 conv layers and 6 pooling layers (total 15 as shown below). The last 2 layers shown in the diagram are used for grid creation, bounding box and class predictions. For our project, we modified the last Conv and Regions layers to have 2 classes (person and pets) and corresponding filters to 35 (2+5 times 5 bounding box). YOLO outputs a confidence score on how certain it is that the predicted bounding box encloses an object. For each bounding box, the cell also predicts a class (ex. person). The two scores are combined to give a probability score.



```
Layer          kernel  stride  output shape
---------------------------------------------
Input                          (416, 416, 3)
Convolution     3×3      1     (416, 416, 16)
MaxPooling      2×2      2     (208, 208, 16)
Convolution     3×3      1     (208, 208, 32)
MaxPooling      2×2      2     (104, 104, 32)
Convolution     3×3      1     (104, 104, 64)
MaxPooling      2×2      2     (52, 52, 64)
Convolution     3×3      1     (52, 52, 128)
MaxPooling      2×2      2     (26, 26, 128)
Convolution     3×3      1     (26, 26, 256)
MaxPooling      2×2      2     (13, 13, 256)
Convolution     3×3      1     (13, 13, 512)
MaxPooling      2×2      1     (13, 13, 512)
Convolution     3×3      1     (13, 13, 1024)
Convolution     3×3      1     (13, 13, 1024)
Convolution     1×1      1     (13, 13, 125)
---------------------------------------------
```

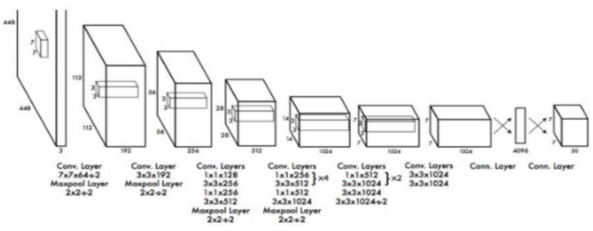


Figure:[20][21] Tiny-YOLO Architecture with 15 layers. We modified the last layer.

Hyper Parameters: The convolution layer calculates: if x is the pixels in the input image and w is the weights for the layer, then the convolution basically computes the following for each output pixel. Then batch normalization is performed with parameters mean, variance, gamma and beta. Our model used a learning rate of 0.00001 and batch size of 16.

```
out[j] = x[i]*w[0] + x[i+1]*w[1] + x[i+2]*w[2] + ... + x[i+k]*w[k] + b
```

```
                gamma * (out[j] - mean)
bn[j] = ------------------------ + beta
                  sqrt(variance)
```

Model#2 - Face Detection with YOLOv3 on WIDERFace Dataset:

In Comparison with YOLOv2, the newer version, YOLOv3 has made several updates on the network architectures and added more layers using the concepts introduced by Resnet and so on. In addition to that, it also runs the detection as three different scales.

YOLOv3 Architecture: YOLOv3 follows the mechanism described in the YOLO9000 for object detection, divides the input image into small regions and predicts the bounding boxes as well as the probabilities for that region and YOLO removes those predictions with low probabilities using non-maximum suppression. YOLOv3 has increased number of layers to 106 as shown below [11][12].

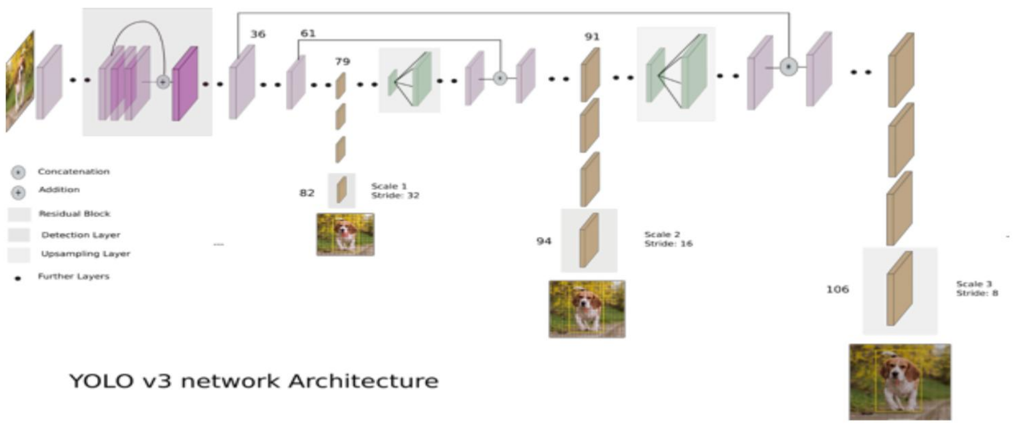


Figure:[11][12] YOLOv3 architecture with 106-layers.

There are also some variants of the networks such as YOLOv3-Tiny and so, which uses less computation power to train and detect with a lower mAP of 0.50. YOLOv3 consist of 3 scales output at layer 82, 94 and 106. It extracts features from those 3 scales using a similar concept to feature pyramid network. The last output layer predicts a 3D tensor with bounding box, object-ness, and class predictions. Those scaled detection layers are given by 1 * 1 * (B * (5 + C)), where B is the total bounding boxes (which is 5). C is the number of classes and it is 1 in our model (i.e. just face).

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

Loss function: [21] is hard to understand as it was not documented well in the YOLOV3 paper. The loss function shown here is used to correct the center and the bounding box of each prediction from the YOLOV1's paper and here we try to describe our own best understanding of the loss function. There are 3 parts in this equation: bounding box, objectness and class predictions.

Hyper Parameters: YOLOv3 has adopted a lot of new improvements such as batch normalization, data augmentation and so on to improve its performance on object detection. For data augmentation, it has 4 hyper Parameters, angle, saturation, exposure and hue. Momentum and Learning rate, and batch normalization are also hyper parameters (which we configured for our model).

Model#3 - Age and Gender Prediction:

Initially, the model used for age/gender prediction is based on the Levi/Hassener's paper [13][14] whose architecture is shown in the figure below (left). It has 5 layers (3 Con+2 FC).

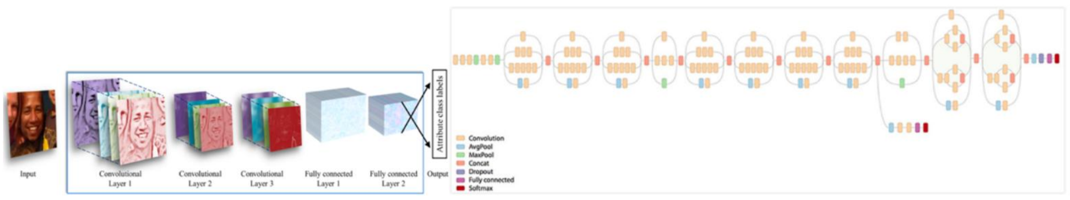


Figure:[13][14] Left shows original Levi's model with 5 layers. Right shows Inception v3 model with 48 layers.

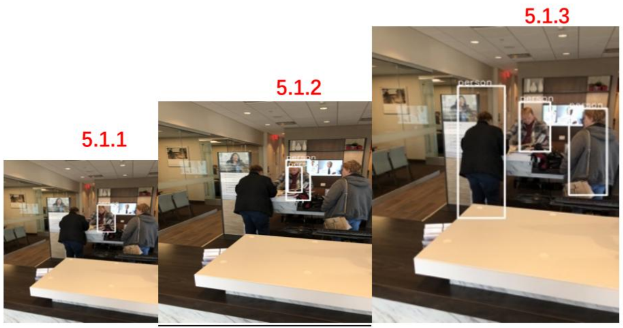
Later, a better result is obtained through transfer learning against Google's inception v3 model [15], which has been pre-trained with over a million images and whose output layer is revised to 2048x8 for age and 2048x2 for gender. We configured both 2D convolution layer and fully connected layer. The architecture for inception v3 is shown above (right). Since the softmax is chosen here for classifying age/gender, both the models have the cross entropy mean with a regularization term as the loss function.

5. Experiments / Result / Discussions

We integrated our project with 3 models to have YOLO v2 just identify person, YOLO v3 model to detect face and CNN model to classify them in gender / age groups.

Model #1 - Person identification with YOLO9000/v2:

We ran the model with pre-trained weights initially and tried several iterations to improve the results. The table below shows 3 main results that improved the accuracy to 0.47. We are not able to surpass 0.57 baseline from YOLO paper due to limited CPU resources, longer training time and average moving loss still over 1. As we can in the pictures, the bounding box prediction improved dramatically with accuracy. In the first case, we tried to mix our pictures with training PASCAL set (and vice-versa) but the model did not converge due to hardware limitation and difference in quality of images.



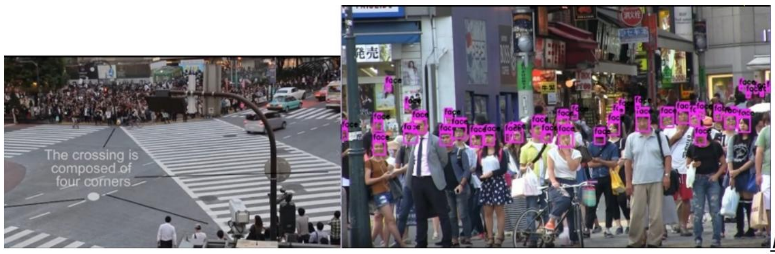
Figures: Pic on right has b-box on all 3 persons. Table shows 0.47 max

Model	Iteration / Epoch	Avg Mov Loss	Accuracy
5.1.1. PASCAL 5000 images + our data 100 pic	9000 / 50	1.925	0.253
5.1.2. JPG to all PNG	5000 / 92	2.725	0.35
5.1.3. Val set expanded to 300	6250 / 71	2.5	0.47

Model#2 – Face Detection Training on WIDERFace Dataset:

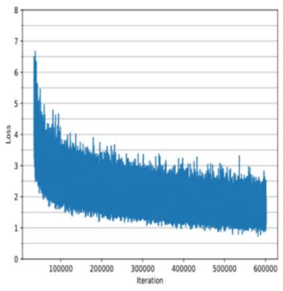
We start the first iteration by using the pre-trained weights, which was trained on Pascal Dataset to get hands on experiences. Due to improvements in productivity features like bounding box, we decided to use AlexeyDB's [16] implementation of YOLOv3 as our training framework. The training on WIDERFace requires modification of the dataset label format and the bounding box sizes. In this paper, with height and width both equal to 608 pixels, we used bounding boxes with the following sizes:

4,7 7,13 12,22 20,36 32,56 55,87 90,148 159,232 268,353



Pictures: Left one has really small faces and cannot be deducted due to bounding box set up. Picture on right has b-box on all faces.

After a few hundred hours of training, we could see improved and converging dev set results as follows. The loss is < 1 near 600k iterations (shown below on left). In this paper, we picked the 500k iterations weights as our final weights since it has the highest mAP@0.50 with 58.78% at 27 ms (Table). Compared YOLO's paper [2] with COCO dataset on 80 classes, the authors achieved 57.90 on mAP@0.50 with inference time 51 ms. Although we are 1 percent higher and much faster than the final YOLOV3-608's COCO weights, this comparison is trivial as running on different dataset, different detection classes and different hardware. However, at least it can tell that the model we trained is performing fairly good though.

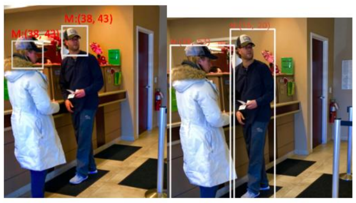


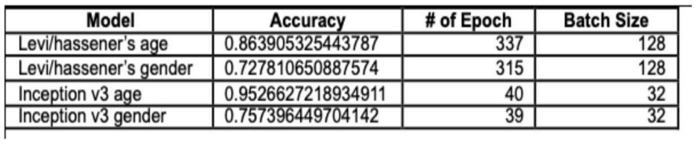
Minibatch steps	Detections count	Unique truth count	AP	Precision Thresh 0.25	Recall Thresh 0.25	F1-score Thresh 0.25	average IoU	mAP @0.50
601K	36629	39708	52.73%	0.92	0.46	0.62	69.39%	52.73%
600K	29910	39708	58.78%	0.95	0.38	0.54	72.68%	50.79%
500K	38959	39708	50.79%	0.91	0.49	0.64	69.27%	58.78%
400K	30327	39708	51.46%	0.95	0.38	0.55	72.54%	51.46%
300K	71187	39708	57.98%	0.88	0.5	0.64	66.72%	57.98%
200K	56827	39708	56.39%	0.92	0.43	0.59	70.52%	56.39%
100K	69728	39708	57.51%	0.95	0.38	0.54	72.33%	57.51%
50K	269399	39708	54.02%	0.81	0.46	0.59	58.33%	54.02%

Figures: Table has loss function converging < 1 (y-axis) for 600K iterations (x-axis). Table shows our model's iterations with max. 58.78%.

Model#3 - Training on Age and Gender Dataset:

For age and gender classification, we trained both Levi and Inception models as explained in previous section, with latter achieving a higher accuracy. Though the models have not seen our test data and despite limited CPU and training time, the transfer learning on inception v3 outperformed paper's accuracy. If we used a full body object (entire person) on our model, the performance of the gender/age classification model is not very accurate as Audience database has only face images. For example, the picture on right has whole person in the box and our model predicted the age to be in 15-20 category. By manually cropping the face images (picture on left), model computed this person to be in age group of 38-43 with 99% accuracy, which is more reasonable. However, the female is detected as "Male" in both pictures because her face is not facing the camera.





Model	Accuracy	# of Epoch	Batch Size
Levi/hassener's age	0.863905325443787	337	128
Levi/hassener's gender	0.727810650887574	315	128
Inception v3 age	0.9526627218934911	40	32
Inception v3 gender	0.757396449704142	39	32

Figure: Picture on left is face-cropped manually, so male classified in age (38-43) but the picture on right has the whole body with less accurate age prediction (15-20). Table shows our model results.

6. Conclusion / Future Work

Coding / GIT [19]: We cloned a lot of model from GIT but wrote a good amount of code for each of the 3 models. Here are the links for 3 models: YOLOv2-Identification, YOLOv3-Face and CNN-Age/Gender.

We are able to achieve an end to end objective of people detection and classification with a reasonable accuracy. Here are some activities we are planning to do as next steps:
Improve accuracy: The accuracy was not above the benchmark set by all three models, but we have created a strong baseline model that supports our end to end use-case. We are planning to have access to GPU, better validation set and alignment of training and our datasets.
Live streaming: We plan to expand with a pipeline which could periodically capture the frame in the video and crop the person's face area (if visible clearly) for each detected person automatically. The coordinates of each detected person/face can be embedded in the file name for backtracking purpose.
Industry Links: We will continue to have industry links to implement a good product for banks. Our article on "AI for Banks" was chosen to be the cover story of a premier banking journal RMA (Risk Management Association) [17] in its March publication (print and digital). We also delivered a keynote address on this topic in a Banking Cyber Security Conference [18] in March.

7. Contributions

We are a team of three SCPD (online) students in different time zones and countries. Yan (who lives in China) was responsible for testing YOLOv3 model. Chu-Chi (who lives in California) was responsible for working with CNN model to classify gender and age. Senthil (who lives in Pennsylvania) was responsible for working with YOLO9000 model and initial preparation of the Validation/Test dataset images.

We want to thank our project mentor Hojat Ghorbani for his guidance and advice throughout the project. We also want to thank all other teaching resources associated with CS230 for guiding us throughout the course.

We want to recognize Upper St. Clair High School at Pittsburgh, PA for its technology programs and one of its students Vidhur V Senthil (son of one of the authors) who helped with annotating the pictures for our project.

8. References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788

[2] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018.

[3] Pascal VOC Dataset. URL: http://host.robots.ox.ac.uk/pascal/VOC/

[4] Age and Gender Classification. Published in IEEE on Computer Vision and Pattern Recognition (CVPR). URL: https://talhassner.github.io/home/publication/2015_CVPR

[5] Tal Hassner, Shai Harel*, Eran Paz* and Roee Enbar, Effective Face Frontalization in Unconstrained Images, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015.

[6] Tzutalin LabelIMG. URL: https://github.com/tzutalin/labelImg

[7]Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger. URL: https://pjreddie.com/media/files/papers/YOLO9000.pdf

[8] Yang, Shuo et al. “WIDER FACE: A Face Detection Benchmark” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[9] Levi/Hessener split train / test dataset – Link

[11] Source: Ayoosh Kathuria. What’s new in YOLO v3?. URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[12] Ayoosh Kathuria “What’s new in YOLO v3?” URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[13] Gil Levi and Tal Hassner. Age and Gender Classification using Convolutional Neural Networks. Department of Mathematics and Computer Science, The Open University of Israel. 20xx Link

[14] Age Gender in GIT. URL: https://github.com/dpressel/rude-carnie

[15] Google-Inception Architecture – Link

[16] AlexeyAB, URL：https://github.com/AlexeyAB/darknet

[17] Risk Management Association (RMA) Banking Journal and AI Cover Story March 2019 – Link

[18] InfoSec Connect, Banking Risk Conference, San Diego CA March 2019 – Link

[20] Tiny YOLO Architecture and layers figures – Link

[21] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4

[22]Konstantine-Buhler-et-al. ”YoloFlow Real-time Object Tracking in Video”. URL: http://cs229.stanford.edu/proj2016/report/BuhlerLambertVilim-CS229FinalProjectReport.pdf