A Deep-Learning Based Approach to Bangla Currency Recognition in Real Time

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering of the University of Asia Pacific

Submitted by

Shakhawat Anol
Roll: 17201017
Ashik - E – Elahi
Roll: 17201028
Imtiyaz Hassan
Roll: 17201033

Supervised by

Md. Hasan Murad
Lecturer
Department of Computer Science and Engineering
University of Asia Pacific

Department of Computer Science & Engineering
University of Asia Pacific

11th December 2021

DECLARATION

We hereby proclaim that the work presented in this project is the result of the analysis conducted by us under the observation of Md. Hasan Murad, Lecturer, Department of Computer Science and Engineering, University of Asia Pacific. We also declare that no part of this project has been or is being submitted elsewhere for award of any degree or diploma.

Signature of Supervisor

হাসানমুরাদ

Supervisor

Md. Hasan Murad

Lecturer

Dept. of CSE

University of Asia Pacific

Signature of Candidate

Shakhawat Anol

Shakhawat Anol

Ashik

Ashik-E-Elahi

Imtiyaz Hassan

Imtiyaz Hassan

Acknowledgement

First, we would like to thank the almighty Allah for giving us the inspiration, skill, patience and the ability to finish the work.

We also would like to thank our honorable supervisor Md. Hasan Murad, my respected supervisor for his proper guidance, support and his Patience. His guidance has made our work easier and more accurate. He gives us the most effective way to solve a problem we faced during propagating this project. His productive criticism helped us to find the proper path and finish our project.

Finally, we would like to thank our all-honorable faculty members, family and friends for their support and being always with us.

Abstract

Nowadays, researchers are trying to provide working solutions to the day-to-day problems faced by visually impaired people. Currency recognition using android phones can be a promising solution for visually impaired people for the safety and security during financial transactions. In this research work, we are going to implement a camera-based system that can detect Bangladeshi currency automatically. Our goal was to use 2500 images to make the dataset. To design our model architecture, we are deploying the state-of-the-art object detector model named YOLOv4. We will train our model using our Bangla currency dataset.

Finally, we used 900 images to train our data model. Our model outcome is satisfactory which encourages us to conduct further research on this particular topic. If we input a video sample of a Bangladeshi banknote in our model, our code gives output in which we can see a video labeled a particular note.

Our project is helpful for visually impaired people. In future we will try to make an android and iOS application so that visually impaired people can use this for their benefit.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This system works with real time currency detection. Basically, it's for visually impaired people who have been deprived from society. According to World Health Organization (WHO) [2] reports that 253 million people are visually impaired globally and among them 36 million are blind and 217 million have moderate to severe vision impairment. From the recent statistics Bangladesh has 650,000 blind adults aged 30 and over in Bangladesh [3]. This system can recognize, identify and show real time results of the currency note of People's Republic of Bangladesh (TAKA). As nowadays transactions of Currency are increased to a large quantity and most of them happen face to face, it is difficult for visually impaired people to know how much currency they are transiting.

This system is based on deep learning methodology. we develop a system that will be able to recognize currency with great accuracy. For this YoLo-v4 will be the best solution. It is faster than any other object detection algorithm.

1.1 Motivation

Our primary motivation was visually impaired people. There are so many people in our country who are totally blind or partially sighted. Some of them are blind by birth, some of them lost their sight due to accidents or injuries. According to research [1] 50–80% of blindness in this region are due to cataracts. Latest research of the World Health Organization (WHO) [2] reports that 253 million people are visually impaired globally and among them 36 million are blind and 217 million have moderate to severe vision impairment. From current estimation, Bangladesh has 6,50,000 blind adults aged with 30 years or more in Bangladesh, the majority of them are agonized from operable cataract [3].

While leading their daily life in society, visually challenged people face a terrible amount of challenges. This is because so much information is expressed visually. Among so many problems one of the major problems is knowing the value of the currency that they are currently holding in their hand. As nowadays transactions of currency are increased to a large quantity and most of them happen face to face, it is difficult for visually impaired people to know how much currency they are transiting.

They may get fraud by dishonest people. Another problem agonized by a visually impaired person to recognize paper currencies. This is because of nearly similar paper size and texture between banknotes. As credits are the commonly used material in day-to-day life, recognizing the value of the banknotes is a very essential task for them. Therefore, we have put forward a real-time system that will help them to recognize the banknotes and resolve this issue to make visually impaired people feel confidence in the financial dealings, not depending on others.

Only country in the world that prints all denominations of Banknotes in the same size is the USA. Only two countries that use the same colors for all of their various bills are The US and Switzerland. It is safe to say that similar size and colors make it hard for a blind person to recognize the correct bills to make a purchase without any assistance [4]. A sightless person can identify paper currency by folding each denomination in different ways. Although the idea of folding the currency seems a good step, it needs others' help to serialize the bills. The recent banknotes of Bangladesh have three dots for blind humans, which can be effective to check the value of the bill by touching. As the embossing fades away in the banknotes that are in circulation for a long time, It is hard to identify the bank note after several uses [5]. Moreover, in a developing country like Bangladesh, visually impaired people fall victim to deception on a regular basis while transacting money. Therefore, to ensure safety and confidence during financial transactions conducted by visually impaired people, they need assistance from technology to recognize Bangla currency. The main argument in this project is realization of deep-learning based currency recognition software. According to the procedure in this project and perform currency recognition step by step, it includes identifying the bank-note.

1.2 Social and Ethical Issues

The term “visual impairment” is employed to explain a large number of conditions that have an effect on clarity of sight and field of sight. Visually impaired people have been deprived from society. Day to day life is not that easy for them. One of the daily matters they have to deal with is acknowledging the currency they are using. Though there are some ways imprinted in every Bangladeshi note to recognize the value just using our touch senses. But not everyone knows about it and it is hard to remember. Also, the note can be damaged in a certain way, it can’t be acknowledged. Another Ethical issue sightless people have to face is getting defrauded by some ill-intentioned people. It is a matter of mortification for our society. So, our approach on this issue using deep-learning algorithms can have a huge impact on society.

1.3 Sustainability Issues

At this moment there are so many systems or applications to detect and identify different types of currency. Several research happens to resolve this problem. The USA Department of Treasury invented a device that can detect currency. There are also some other apps available and these systems are able to give output in real time. But for Bangladeshi currency there is no well-known, easy to use system. This gives us an opportunity to provide a sustainable system that can detect Bangladeshi currency in real time. A large number of visually impaired humans face hassle in their daily lives during currency transactions.

Our model is based on Bangladeshi banknotes. We trained our model by providing bank currency as a dataset. We provided all present banknotes in our system that are circulated by Bangladesh Bank. Our model’s sustainability depends on our dataset. We ran our model for nearly four months and the result is satisfactory. This currency detection model is sustained as long as our dataset is valid in our country. We know Bangladesh Bank circulates new bank notes after several time periods. So, until new banknotes get in circulation, our model can perform well. We are glad to declare that we can update our model if we update our dataset. That means if authorities issue new banknotes and if we train our model with these new notes our model will become sustainable for those datasets as well.

1.4 Related Work and Limitation

For this project, this report will read many research papers on currency recognition, object detection, image processing and edge detection (ORB). We tried to refer them into the reference section of this report. But we evaluate some projects and research papers on our own.

The papers we are trying to uphold here are :- Prof. Sagar S.Rajebhosale [7], Sandeep Kumar Chaubey [8] and Md. Ferdousur Rahman Sarker [9]. All of these systems are well proposed and designed. But we found some relative issues that we are going to explain in a later segment of this report. And the project we are developing will overcome all these problems.

1.5 Problem Statement

An approach based on deep learning methodology to solve the currency recognition problem for the people suffering from visual impairment. A system that can recognize, identify and show real time results of the currency note of People's Republic of Bangladesh (TAKA). In this project we are also developing skills on image annotation, image processing, object detection model training.

1.6 Contribution

Here we have created a model to recognize Bangladeshi currency. Our model can identify all running banknotes that exist in Bangladesh. Here we are going to summarize our contribution.

In the past, no perfect dataset has worked on recognizing Bangladeshi currency. We worked on the dataset to label it correctly for training purposes. Training purpose was not 100% accurate. We tried to train it to achieve more accuracy.

We used YoLo v4 to train our dataset. Currently it is the best option for object recognition in machine learning. We have done training, testing, validation for our dataset and model. Finally, we write this paper based on our project.

Chapter 2

Background Study

In background study we will talk about the knowledge we had to gather to complete this project. We will discuss object detection, machine learning, deep learning etc.

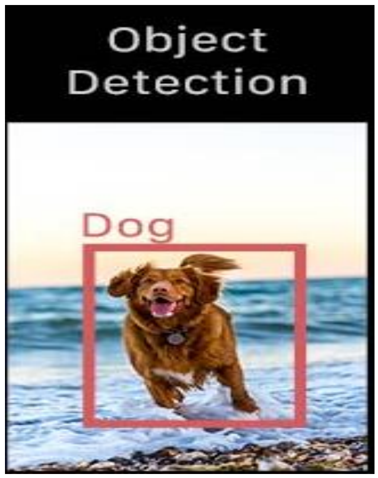

Figure 2.1 : Object Detection

2.1 Object Detection

Object detection allows us to identify and locate an object from an image. This technology is related to computer vision and image processing which work with semantic object class. In short, if we want to define what object detection is then we can say this technique is to draw a bow around the object and label the particular object. The model decides which object is and which label is to apply.

2.2 Machine Learning

Machine learning is a study of algorithms. It can improve by itself by experience and use of data. Basically, machine learning is a process of data analysis which helps us to build analytical models in an automated system. In this technology a computer can learn by itself by using data.

2.2.1 Machine learning Approach

Supervised Learning

Supervised machine learning is a subclass of machine learning. Basically, it uses a labeled dataset to train algorithms that we use for determining objects. As input data is provided into the model. This model is being continuously adjusted until the weight of the model is well fitted, which occurs as part of the go through validation process.

Unsupervised Learning

Unsupervised learning algorithm use for unlabeled and non-classified data sets. This algorithm couldn’t find out target output but draw inference from datasets that people aren’t explicitly looking for. Example of unsupervised learning clustering algorithm, dimensionality reduction.

Reinforcement Learning

Reinforcement learning (RL) may be an important sector of machine learning. It deals with how an intelligent agent takes action to support the thing. It doesn't like any input or output labels. It abides by the Andrei Markov call method. The purpose of reinforcement learning is to show a man-made agent to require action by gazing at object or associate surroundings. Here the bogus agent is trained o.k. in order that the agent will take the correct steps at the correct time.

2.3 Deep Learning

Deep Learning is a subclass of machine learning apprehensive with algorithms inspired by the structure and performance of the brain called AI. Deep learning AI is in a position to find out without human interval or supervision, drawing from a dataset that's both structured and unmarked. Deep learning uses multiple layers to extract complex high-level information.

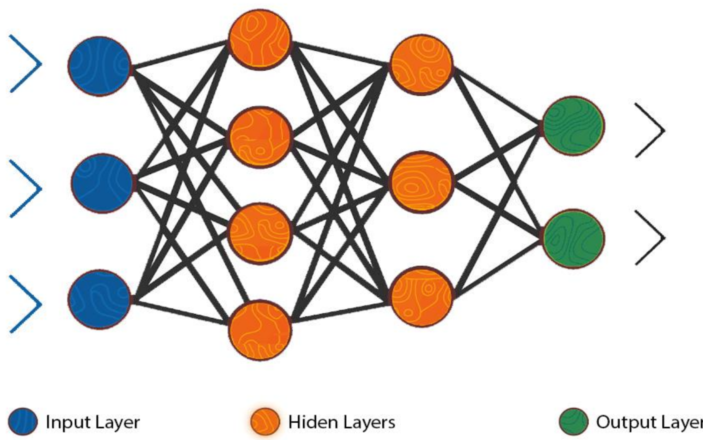

Figure 2.2 : Hidden Layers

But we cannot identify a neural network as a deep learning until there is total three layers. These layers can be hidden layers or input-output layers. In the figure above, we can see 4 layers, two hidden layers in the middle surrounded by one input and one output layer. This neural network can be called deep learning.

2.4 Model Architecture

Different model architectures are used in machine learning for different purposes. One is used to classify images, one is good for predicting the next item in a sequence, and one is good for sorting data into groups. Some are good for multiple purposes, and some are good for just one.

2.5 Computer Vision

Computer vision is a subject where a computer discusses how to acquire a high level of knowledge from a picture or video. It can do a whole job quickly and very easily. Moreover, it can repeat a task very easily. Computer vision always tries to give very perfect results. It always plays an important role for face recognition. Here are some of the important sub-area of computer vision, Image classification, Object recognition, Object tracking.

2.6 Image Classification

Image Classification is a central assignment that endeavors to appreciate a whole picture overall. The objective is to arrange the picture by doling it out to a particular mark. Regularly, Image Classification alludes to images in which just one item shows up and is dissected. Conversely, object discovery includes both characterization and confinement errands, and is utilized to break down more sensible cases in which different items may exist in an image.

2.7 Object Recognition

Object Recognition is a sub-area of computer vision. Object Recognition is a technique to identify objects in images or videos. There are a large number of practical applications of Object Recognition. Face Recognition, Face Detection, Image Annotation, Activity Recognition, Video object co-segmentation are some examples of object recognition.

2.8 Object Trucking

Object Tracking algorithms detect moving objects in videos. Traffic controlling, medical imaging, sports analysis, autonomous driving, security and surveillance are most common object tracking work.

Chapter 3

Related Work

3.1 Paper review

We search for research papers, projects and live software on the internet. We got so many results and trying to uphold few of them here,

3.1.1 Paper “Currency Recognition System Using Image Processing” :

A research paper called “Currency Recognition System Using Image Processing” was published by Prof. Sagar S.Rajebhosale [7]. In this paper the researchers proposed a system based on image processing. In this method input Image will be converted into digital form and then compared with the previously setup template. They propose to use a different system than the user. Users have to upload the image to a server, then the server will process it and return the result to the user.

3.1.2 Paper “Currency Recognition System using image processing” :

Sandeep Kumar Chaubey [8] submitted a report on “Currency Recognition System using image processing”. In this paper the researcher uses image analysis, image processing and edge detection. Image analysis tasks such as reading bar-code tags or as sophisticated as identifying a person from their face. Edge detection is used in image analysis for recognizing region boundaries. Researchers are using the MATLAB environment. MATLAB is considered as a high-level advanced Computing language. It has a variety of API for image processing.

3.1.3 Paper “Real-Time Bangladeshi Currency Detection System for visually impaired persons” :

“Real-Time Bangladeshi Currency Detection System for visually impaired persons' ' named research paper was published by Md. Ferdousur Rahman Sarker [9]. In this paper, the researchers propose an image processing based system that will use Keypoints Detection. Their approach to the system is very impressive. The proposed system can be reliable and process time for each image will be fast because of ORB Keypoint detection.

3.2 Limitations :

Now we will try to hold up the limitations of these system we reviewed in 3.1 . At first Paper of Prof. Sagar S.Rajebhosale titled “Currency Recognition System Using Image Processing”. This system is not fast enough, like, the total time for returning the result might take a while depending on the multiple variables. For example, users have to upload the image to the server then the server will process the image with various operations, compare with the database in the server and return the result to the user. In our observation, this is the main limitation of this system. It’s not reliable because the main processing is not happening on the user's device.

Second paper we reviewed above was published by Sandeep Kumar Chaubey, titled “Currency Recognition System using image processing”. After analyzing their research paper we found some limitations: they use image processing , image analysis and edge detection which don’t get a high percentage of accurate results. They use the MATLAB environment. Matlab is very complicated to use. Matlab is a confusing mixture of multiple programming languages like c, c++ and java. So manipulating this system is quite complicated in our observation.

The last paper was titled “Real-Time Bangladeshi Currency Detection System for visually impaired persons”, published by Md. Ferdousur Rahman Sarker. After reading about their paper and researching their approach, we have found some slight limitations on this method. Though they train the neural network with ideal and distorted images, it still might have some issues. Though they used Keypoint Detection to make it fast ,but when the bank-note is folded Artificial Intelligence won’t be able to find all the Keypoints. Hence, the result accuracy will decrease. Similar problems also can occur when the bank-note is stained.

Chapter4

Proposed Solution

In this phase, we discussed about the model architecture of YoLo-V4 model and details about out model training: -

4.1 Model Architecture: YoLo V4

YoLo was originally authored by Joseph Redmon to detect objects. The full meaning of Yolo is “ You Only Look Once”. YoLo is a sophisticated Neural Network for object detection in real time. In YoLo, a single neural network predicts multiple boundary boxes simultaneously and class probabilities for those boxes.YoLo works on complete images and directly optimizes detection performance and speed.

YoLo-v4 model published in April 2020 and achieved state of the art performance in COCO dataset. The object detection task broken into two steps :

- Regression to identify object positioning through bonding boxes
- Classification to identify the object classes

Yolo-v4 outran all existing methods significantly in both terms detection performance and superior speed. The main objective is to optimize neural network detectors for parallel computation. Yolo-v4 consist of :

1. Backbone (CSPDarknet53)
2. Neck (Spatial Pyramid Pooling, PANet path aggregation)
3. Head (YoLo-v3)

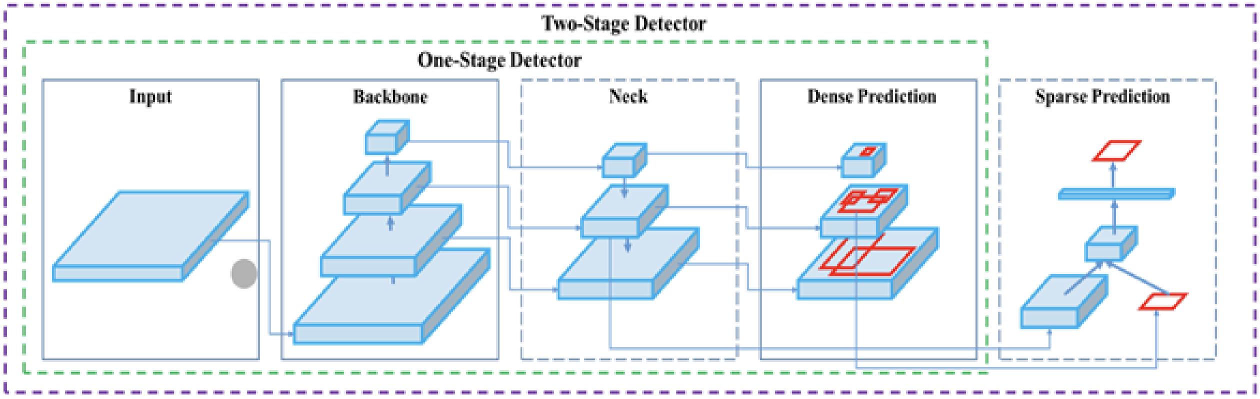

Figure 4.1 : Stages of Yolo-v4

4.2 Model Training

Model is one of the most crucial parts of this project. A Neural Network Model is a technique for approximation of an unknown function using a dataset or observation. In another construct, a neural network model resembles the way a human brain works. Yolo-V4 model training has some different steps then all other Neural network training methods. To train our model we had to create our own dataset including 9 classes. Using Label-Img software, we add annotation in Yolo format. We used a pretrained weight (yolov4-tiny.conv.29) to train our model weights. In this project we used several tools :-

1. Label-Img Master
2. Anaconda
3. Notepad++
4. GitHub Desktop
5. Google Co-lab
6. Window Photo Editor

Chapter 5

Dataset

The dataset we used was a creation of our own. This type of project is not very common, especially the currency we are working on. We didn't find any open dataset that can fulfill our project outcome. So, we create our own dataset, collecting different types of credits. For all nine classes, we primarily collected 200 individual data for each corresponding class. The initial dataset was 1800 data. Unfortunately for limited resources, we had to size down the dataset to 900 (100 data for each class).

5.1 Classes

In Yolo-v4, to train the model to detect object from still or motion picture we have to identify these objects as classes. This is a very important part of the model design and training. Because the classes we input from the beginning will be in the final model. And iteration number of the train is also depended on how many classes are present in the model. Here is the class list for this project:

1. Two_taka	6. One_hundred_taka
2. Five_taka	7.Two_hundred_taka
3. Ten_taka	8.Five_hundred_taka
4. Twenty_Taka	9. Thousand_taka
5. Fifty_taka	

5.2 Data Processing

There were two stages in data preprocessing. First one was re-size all images and lower resolution. Second stages were dataset labeling.

5.2.1 Re-size data :

The initially collected images was 4000*1900 pixels resolution. We used a tool call Windows Photo Edit to size down these images. After processing the ready image's resolution was 1274*720 pixels. It also brought down the image size to 110 to 170 KB. This part was very crucial for data training because high resolution images use more resource to train.

5.2.2 Dataset Labeling :

To help the model to identify object in a image, these images had to be labeled. Which is called data annotation. We used a software known as Label-Img Master and with the help of anaconda programming language to label these images in designated classes. Data annotation differs between model to model. The format we use was yolo format which generate a txt file containing number of the class and X1 Y1 , X2 Y2 co-ordinates of marked image. Here is a example of Ten taka data labeling

Figure 5.1 : Data Annotation

5.3 Dataset Example :

Figure 5.2 : Dataset

Chapter 6

Result

We configured our model to train up to 7000 iterations. Here is the corresponding loss vs iteration and mAP vs iteration graph. In the Loss vs Iteration graph, we can see that with increment of iterations, reduction of loss. At 7000 iteration loss becomes 0.0206.

Figure 6.1: Model loss graph

On the other hand, mAP vs iteration graph, with the iteration increases , mean average precision (mAP) increases. At the maximum iteration our mAP was 22%. There is room for improvement in this attribute, but it demands more resources.

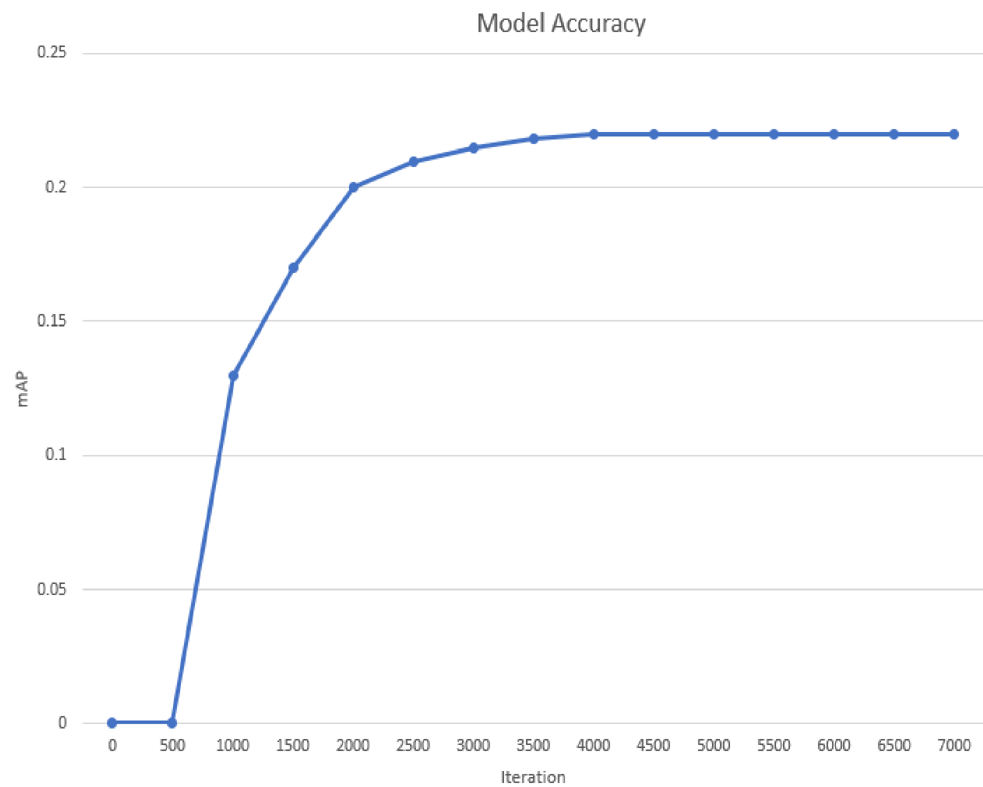

Figure 6.2: Model Accuracy graph

6.1 Training Graph

This graph was automatically generated by the framework called darknet. It resembles the whole training process. mAP, initial loss, last loss, current average loss, iteration, time to complete and max_batch, all of these data is shown in the graph below.

In this 2-axis graph, x-axis represents the loss of every iteration and y-axis represents the value of iterations. We can see the initial loss of this model was really high. This initial loss drastically drops between 0 to 100 iterations. After 700 iterations the loss is already below 1. From 2100 iterations loss drops very small amounts at a time. At this point loss might be very low still the accuracy of this model is not good at all. So, we need to complete all 7000 thousand iterations to improve accuracy. After completing the training our average loss was 0.0206.

This graph represents another important attribute called Mean Average Precision also known as mAP. In object detection, mAP is an evaluation matrix. Using mAP in a deep learning model, we can determine the percentage of successful detection for every thousand iterations. mAP increases proportionally with iteration increases.

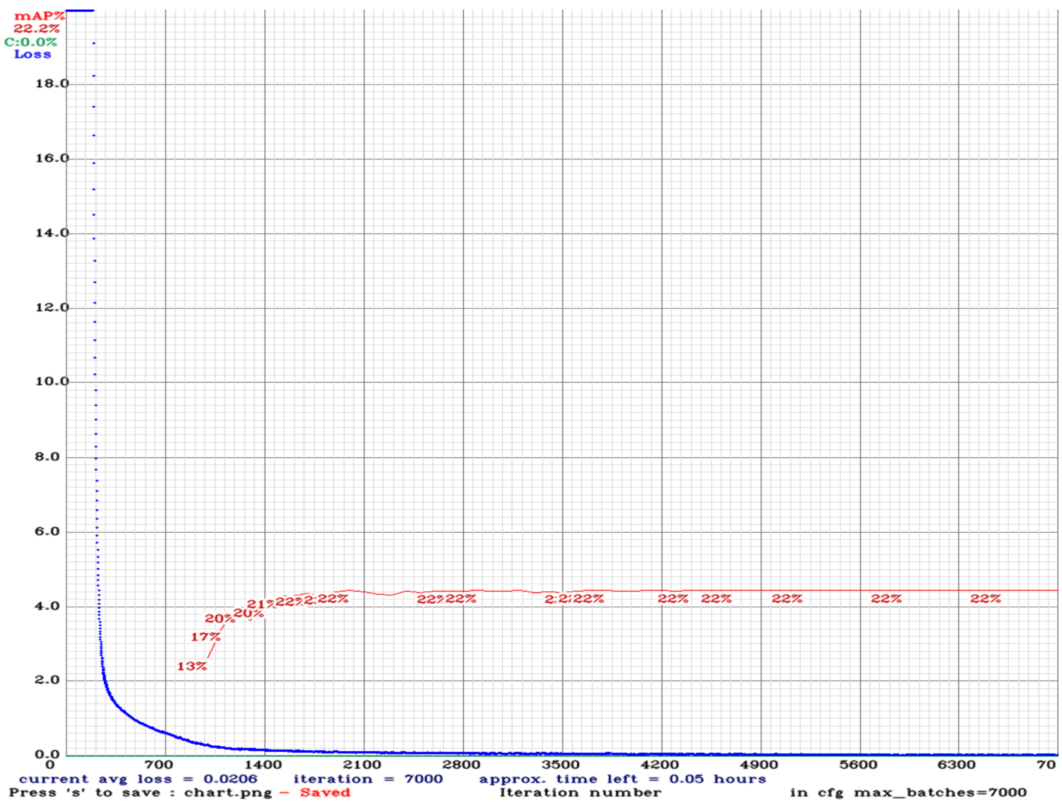

Figure 6.3 : Training Chart

6.2 Evaluation

After completing ours model training it was time to run some test. We prepare some motion pictures to evaluate our model and we were successful. After uploading these videos, we used darknet framework to run some code in Google Co-Lab to input the video and get a output video containing example of detection. We cannot attached these video in a pdf or doc file so we took screenshots for each of the banknotes from these videos. Here is the result of our custom trained model of Bangla currency detection. All the 9 classes of objects are presented below: -

Two taka result

Figure 6.4 : Two taka detection

Five taka result

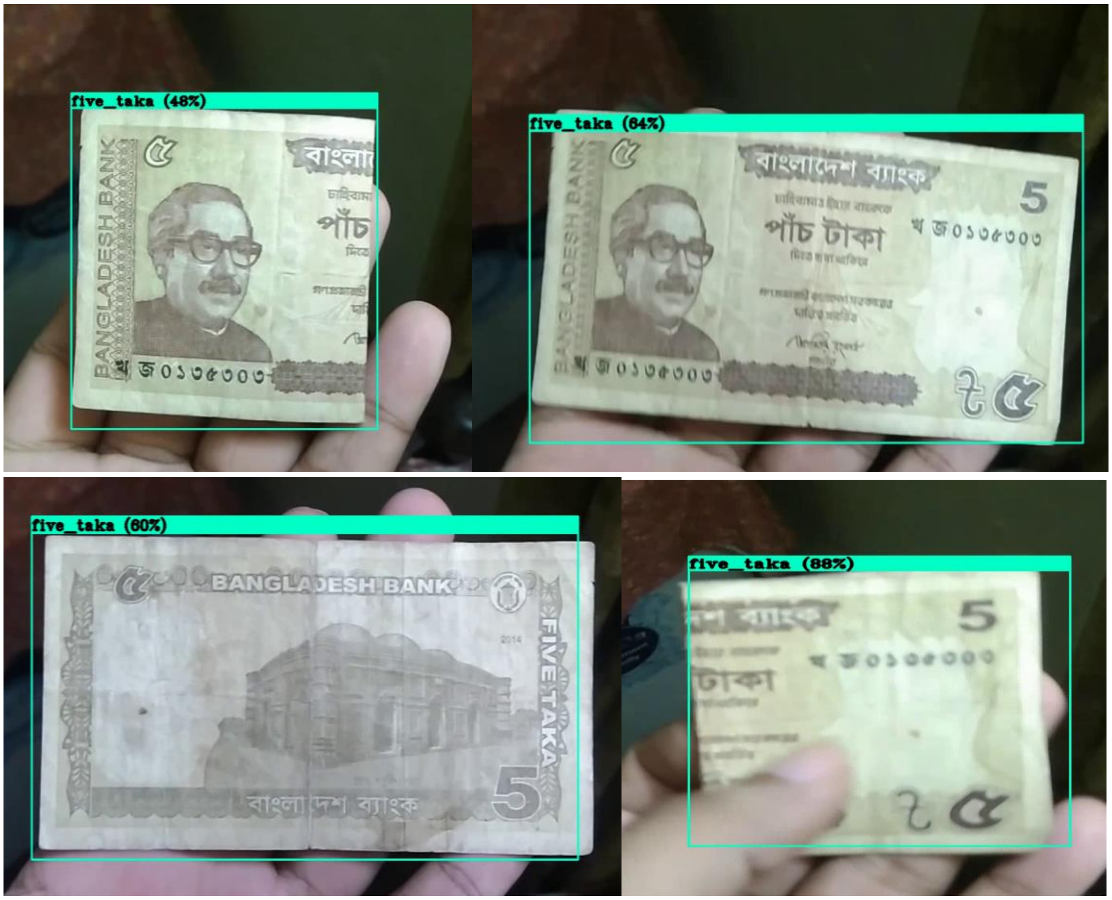

Figure 6.5 : Five taka detection

Ten taka result

Figure 6.6 : Ten taka detection

Twenty taka result

Figure 6.7 : Twenty Taka

Fifty taka result

Figure 6.8 : Fifty taka detection

One hundred taka result

Figure 6.9 : One hundred Taka detection

Two hundred taka result

Figure 6.10 : Two hundred Taka detection

Five hundred taka result

Figure 6.11 : Five hundred taka

One thousand taka result

Figure 6.12 : One thousand taka detection

6.3 Summary of Finding

In this part of this report we like to discuss our findings and the problems we faced during all the steps of our project.

Image Difference : In the data collection process, we (all 3 members) individually collect pictures of different bank notes and also ask some of our friends to help us with this matter. But we faced issues in our model training phase. The problem was, not all capture devices are the same , they have different sensors , different image processing algorithms. So, output from all these devices were different like resolution, size, color accuracy, lighting condition. We were able to fix the resolution and size, but because of different color accuracy our model had a hard time detecting different bank notes. Also different bank notes have nearly the same color(Example: the newly released 50 taka and 200 taka bank notes).

Bank Notes Design : Bank notes design deferred for different values but all of them have something in common. Now a days nearly all bank notes of Bangladesh has a portrait of our Father of Nation Bongo Bondhu Sheikh Mujibur Rahman alone with the printed text “Bangladesh Bank” , “বাংলাদেশব্যাংক”. This type of similarity causes some confusion in the trained model.

Another issue was the condition of the bank notes. For staying in circulation some notes condition becomes very poor because of dirt and some cases because of abuse. These poor condition notes are hard to detect or even make conflict when trained in a model.

Resource Limitation : Some of the limitations we discussed earlier could be resolved (like the similarity and poor condition) with larger and accurate image dataset. For argument sake, let’s pretend we have a larger dataset with multiple types of bank note models. But we could utilize larger data because of hard resource limitations. Larger data set for this many classes, needs more graphics memory to train as well as take a lot of time to train a model. We have done all of our training, test and demo in Google Co-lab. Google Co-lab doesn’t allow run-time more than 12 hours for free users. This was a major setback for this project. Double the amount of dataset was available then we used in training.

Chapter 7

In this chapter, we are discussing the final outcome of this project and our future plan and vast possibility of this project.

7.1 Conclusion

Our paper advanced a project for recognizing currency using object direction from YoLo V4.as YoLo means “you only look once”, so we have a very good theory about this system faster than any other currency recognition software that has been released to the public until now.

We collect Bangladeshi bank notes as data. We collected 2 taka,5 taka, 10 taka, 20 taka, 50 taka, 100 taka, 200 taka, 500 taka and 1000 taka notes to train our model. We labeled our data using Label-Img Master. We ran it using anaconda. After data labeling we train our dataset using YoLo-v4 and test it using a sample image. Though we looked for top most accuracy but couldn’t achieve 100% accuracy. In the future we will make our project more accurate.

We proposed to implement this model in an application for mobile devices such as android and iOS. But we work really hard to collect data and train this model. So, we really could have enough time to complete the application. But this can be done in future.

After implementing all our codes and labeled datasets we insert some sample video. This shows us that our system can identify Bangladeshi banknotes without facing any issue. This is a great achievement for our team and this project is going to help a large number of people who cannot see properly. Our initial goal was to help visually impired people. People who cannot see properly or lost partial visualisation, can get help to recognise bank notes by using our system. We will work further on our project to make it usable for mass people.

7.2 Future Work

We have plans to work on this project in future. We will try to develop android and iOS applications for android and apple users. This will reach our initiative to the people who are visually challenged. We know android has google assistant and iOS has Siri as voice assistant. Visually challenged people as well as blind people can use those voice assistants to open the money recognition app and can identify the banknotes, he/she holds.

So far this is our future plan. Hopefully we can work on this and produce a beneficial output for our society.

References

[1]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1771758/#:~:text=Cataract%20is%20responsible%20for%2050,of%20blindness%20in%20this%20region.&text=Within%20the%20adult%20population%20more,cataract%20in%20Bangladesh%20at%20present.

[2]http://www.who.int/blindness/en/

[3]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1771736/

[4]https://www.afb.org/blindness-and-low-vision/using-technology/accessible-identification-systems-people-who-are-blind-0

[5]https://ieeexplore.ieee.org/abstract/document/9084039

[6] K. K. Debnath, J. K. Ahdikary, M. Shahjahan, "A currency recognition system using negatively correlated neural network ensemble", 2009 12th International Conference on Computer and Information Technology (ICCIT), 21-23 December, 2009.

[7] Prof. Sagar S.Rajebhosale, Devang S.Gujarathi, Sushil V.Nikam, Prathmesh P.Gogte and Nilesh M.Bahiram, "Currency Recognition System Using Image Processing", International Research Journal of Engineering and Technology (IRJET), 03 March, 2017.

[8] Sandeep Kumar Chaubey, "Currency Recognition System using Image Processing", Dinabandhu Andrews institute of technology and management (DAITM), 24 August, 2019.

[9] Md. Ferdousur Rahman Sarker , Md. Israfil Mahmud Raju , Ahmed Al Marouf , Rubaiya Hafiz, Syed Akhter Hossain, "Real-Time Bangladeshi Currency Detection System for visually impaired

persons", International Conference on Bangla Speech and Language Processing (ICBSLP), 27th September, 2019.

[10] G. A. H. Hassanpour, A. Yaseri, “Feature extraction for paper currency recognition,” in In International symposium on signal processing and its applications (ISSPA), Sharjah, UAE, 2007.

[11] Z. Ahmed, S. Yasmin, M. N. Islam, and R. U. Ahmed, “Image processing based feature extraction of bangladeshi banknotes,” in The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), pp. 1–8, Dec 2014.

[12] S. O. A. Ahmadi and T. Kosaka, “Improvement of the reliability of bank note classifier machines,” vol. 2, pp. pp 1313–1316, 2004.

[13] F. Takeda and T. Nishikage, “Multiple kinds of paper currency recognition using neural network and application for euro currency,” in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 2, pp. 143–147 vol.2, 2000.

[14] H. Shen, K.-Y. Chan, J. Coughlan, and J. Brabyn, “A mobile phone system to find crosswalks for visually impaired pedestrians,” Technology and disability, vol. 20, no. 3, pp. 217–224, 2008.

[15] A. Bhowmick and S. M. Hazarika, “An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends,” Journal on Multimodal User Interfaces, vol. 11, no. 2, pp. 149–172, 2017.

Appendix A

How Ks is addressed through the project and mapping among Ps, COs, and Pos

Ks	Attribute	How Ks is addressed through the project	COs	POs
K3	**Engineering Fundamental**	Knowledge of Deep-Learning and good understanding of Darknet Framework is required for this project. Programming language knowledge (python, anaconda) is also needed.	CO1	PO-l
K4	**Engineering Specialist Knowledge**	On this proposed solution, there will be YoLo-v4 model training and testing. This can be a great engineering practice.	CO5	PO-f, PO-h
K5	**Engineering design**	This project shows how we solve modern society engineering problems with an effective approach.	CO4	PO-g
K6	**Engineering practice**	Knowledge about required programming language required for this project.	CO1	PO-l
K8	**Research Literature**	To complete this project properly, we had to conduct research on existing papers or nearly relevant papers on object detection and image processing. We also identify how they solve or propose to solve the problem. We also analyze limitations of their system.	CO1, CO2	PO-l, PO-b, PO-c

How Ps are addressed through the project and mapping

Ps	Attribute	How Ps are addressed through the project	CO	PO
P1	Depth of Knowledge Requirement	Knowledge of software engineering and app developing (K3), (K4) Multi-layer of Engineering model design (K5) Knowledge of engineering practice(k6) In this project we studied deep-learning system using Yolo-v4 model architecture for object detection (K8)	CO2, CO7	PO-a, PO-b, PO-c, PO-d, PO-e,
P2	Range of Conflicting Requirement	Here, we need to make our model fast and accurate but to make it fast we have to use less dataset. Then again to gain higher accuracy we have to use a large dataset.	CO1, CO3	PO-l, PO-k
P3	Depth of Analysis Required	Bangla Currency detection never been done using Yolov4	CO1, CO2	PO-l, PO-b, PO-c
P5	Extent of applicable codes	Collecting data, resizing data, labeling dataset, train and testing model.	CO6, CO7	PO-i, PO-c, PO-d, PO-e

P7	Interdependence	To design this model, we have to collect and select a dataset with a proper image. Selecting appropriate weights is also important.	CO1, CO3, CO6,	PO-l, PO-k, PO-i

How As are addressed through the project

As	**Attribute**	**How As are addressed through the project**
A1	Range of Resources	In the development stage, the project requires are :- Material : Image of all Bank-notes Technology : Yolo-v4, Darknet Framework People : Developers Equipment : Anaconda , Google Co-lab
A2	Level of Interaction	In this project, the biggest issue was to lower the runtime of our Yolo-v4 model training. We solved this issue by re-sizing both our collected images and dataset.
A3	Innovation	The dataset we used to train the model was created completely from scratch.

A4	Consequences for society and the environment	Giving people independence over their life can have a big impact on society. In this project we are trying to empower visually impaired people living among our society.
A5	Familiarity	This project was completed out of familiarity for this team. This is the first time we are using Deep-learning, darknet framework and Yolo-v4.

CO-PO mapping for this project

CO No.	**CO Statements**	**Corresponding POs**
CO1	Identify a real-life problem (**Bangla Currency Detection**) that can be translated to engineering and/or computing solutions by going through design, development, and validation.	PO-l
CO2	Identify results and functional requirements of the proposed solution (**Object detection based on banknotes)** considering software or hardware specification and standards.	PO-b, PO-c
CO3	Identify sub-components of a complex problem, prepare a timetable and proper budget using the project management skills.	PO-k
CO4	Identify and validate the impact of environmental considerations and the sustainability of a system of a complete project .	PO-g

CO5	Assess professional, ethical, and social impacts and responsibilities of the design project .	PO-f, PO-h
CO6	Function effectively in a multidisciplinary team.	PO-i
CO7	Analyze, design, build, and verify engineering systems with given specifications and requirements .	PO-c, PO-d, PO-e
CO8	Present design project results through verbal presentations.	PO-j

Appendix B

Data Processing

```
import os
count = 0

for i in os.listdir():
    os.rename(i,str(count)+ '.'+ i.split('.')[-1])
    count+=1
```

Figure A1 : rename data

Image processing

```
conda install pyqt=5
pip install pyqt5
```

```
C:\Users\ashik\Documents\GitHub\Money-Detection\Data_Collection\Labelimg
conda install -c anaconda lxml
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py C:\Users\ashik\Documents\GitHub\Money-Detection\Data_Collection\custom_data
        C:\Users\ashik\Documents\GitHub\Money-Detection\Data_Collection\custom_data\classes.txt
```

Figure A2 : Label data commands

Model Training

```
from google.colab import drive
drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

Figure A3 : Mount Drive

```
[ ]   1 !unrar x '/content/drive/My Drive/Cu_Detect/custom_data.rar' '/content/drive/My Drive/Cu_Detect/'
```

```
1 !git clone "https://github.com/AlexeyAB/darknet.git" "/content/drive/MyDrive/Cu_Detect/darknet"
```

```
Cloning into '/content/drive/MyDrive/Cu_Detect/darknet'...
remote: Enumerating objects: 15376, done.
remote: Total 15376 (delta 0), reused 0 (delta 0), pack-reused 15376
Receiving objects: 100% (15376/15376), 14.01 MiB | 6.62 MiB/s, done.
Resolving deltas: 100% (10339/10339), done.
Checking out files: 100% (2050/2050), done.
```

```
[ ]   1 %cd "/content/drive/MyDrive/Cu_Detect/darknet"
```

```
/content/drive/MyDrive/Cu_Detect/darknet
```

Figure A4 : Clone Darknet

```
[ ]   1 !make clean
```

```
rm -rf ./obj/image_opencv.o ./obj/http_stream.o ./obj/gemm.o ./obj/utils.o ./obj/dark_cuda.o ./obj/convolutional_layer.o ./obj/list.o ./obj/image.o
```

```
1 !make
```

```
chmod +x *.sh
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv` -DGPU
./src/image_opencv.cpp: In function ‘void draw_detections_cv_v3(void**, detection*, int, float, char**, image**, int, int)’:
./src/image_opencv.cpp:946:23: warning: variable ‘rgb’ set but not used [-Wunused-but-set-variable]
                 float rgb[3];
                       ^~~
./src/image_opencv.cpp: In function ‘void draw_train_loss(char*, void**, int, float, float, int, int, float, int, char*, float, int, int, double)’:
./src/image_opencv.cpp:1147:13: warning: this ‘if’ clause does not guard... [-Wmisleading-indentation]
             if (iteration_old == 0)
             ^~
./src/image_opencv.cpp:1150:10: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the ‘if’
          if (iteration_old != 0){
          ^~
./src/image_opencv.cpp: In function ‘void cv_draw_object(image, float*, int, int, int*, float*, int*, int, char**)’:
./src/image_opencv.cpp:1444:14: warning: unused variable ‘buff’ [-Wunused-variable]
         char buff[100];
              ^~~~
./src/image_opencv.cpp:1420:9: warning: unused variable ‘it_tb_res’ [-Wunused-variable]
     int it_tb_res = cv::createTrackbar(it_trackbar_name, window_name, &it_trackbar_value, 1000);
         ^~~~~~~~~
./src/image_opencv.cpp:1424:9: warning: unused variable ‘lr_tb_res’ [-Wunused-variable]
     int lr_tb_res = cv::createTrackbar(lr_trackbar_name, window_name, &lr_trackbar_value, 20);
         ^~~~~~~~~
./src/image_opencv.cpp:1428:9: warning: unused variable ‘cl_tb_res’ [-Wunused-variable]
     int cl_tb_res = cv::createTrackbar(cl_trackbar_name, window_name, &cl_trackbar_value, classes-1);
         ^~~~~~~~~
./src/image_opencv.cpp:1431:9: warning: unused variable ‘bo_tb_res’ [-Wunused-variable]
```

Figure A5 : Make configuration

```
[ ]  1 %cd "/content/drive/MyDrive/Cu_Detect/"

/content/drive/MyDrive/Cu_Detect
```

```
[ ]  1 !ls darknet/cfg/
```

```
[ ]  1 !darknet/darknet

usage: darknet/darknet <function>
```

```
[ ]  1 !python custom_data/creating-files-data-and-name.py
```

```
[ ]  1 !python custom_data/creating-train-and-test-txt-files.py
```

Figure A6 : Pre-train

```
1 !darknet/darknet detector train custom_data/labelled_data.data darknet/cfg/yolov4-tiny_custom.cfg custom-weight/yolov4-tiny.conv.29 -dont_show -map
 total_bbox = 446843, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.911932), count: 16, class_loss = 0.002935, iou_loss = 0.344748, total_loss = 0.347683
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 446859, rewritten_bbox = 0.000000 %

 (next mAP calculation at 7000 iterations)
 Last accuracy mAP@0.50 = 22.19 %, best = 22.21 %
 6999: 0.011723, 0.022527 avg loss, 0.000010 rate, 1.672547 seconds, 447936 images, 0.048285 hours left
Loaded: 0.000060 seconds
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.913560), count: 16, class_loss = 0.024074, iou_loss = 0.258387, total_loss = 0.282461
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 446875, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.895543), count: 16, class_loss = 0.000815, iou_loss = 0.410283, total_loss = 0.411098
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 446891, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.896744), count: 16, class_loss = 0.001159, iou_loss = 0.338067, total_loss = 0.339226
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 446907, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.925841), count: 16, class_loss = 0.000440, iou_loss = 0.363838, total_loss = 0.364279
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000545, iou_loss = 0.000000, total_loss = 0.000545
 total_bbox = 446923, rewritten_bbox = 0.000000 %

 (next mAP calculation at 7000 iterations)
 Last accuracy mAP@0.50 = 22.19 %, best = 22.21 %
 7000: 0.003581, 0.020633 avg loss, 0.000010 rate, 1.516401 seconds, 448000 images, 0.047807 hours left

 calculation mAP (mean average precision)...
 Detection layer: 30 - type = 28
 Detection layer: 37 - type = 28
136
 detections_count = 383, unique_truth_count = 133
class_id = 0, name = two_taka, ap = 0.00%        (TP = 0, FP = 0)
class_id = 1, name = five_taka, ap = 99.71%      (TP = 51, FP = 9)
class_id = 2, name = ten_taka, ap = 0.00%        (TP = 0, FP = 2)
class_id = 3, name = twenty_taka, ap = 0.00%     (TP = 0, FP = 0)
class_id = 4, name = fifty_taka, ap = 0.00%      (TP = 0, FP = 4)
```

Figure A7 : training

Model Map

```
1 !darknet/darknet detector map custom_data/labelled_data.data darknet/cfg/yolov4-tiny_custom.cfg "backup/yolov4-tiny_custom_last.weights" -points 0
```

```
  17 max                2x 2/ 2     52 x  52 x 256 ->   26 x  26 x 256 0.001 BF
  18 conv     256       3 x 3/ 1     26 x  26 x 256 ->   26 x  26 x 256 0.797 BF
  19 route  18                                  1/2 ->   26 x  26 x 128
  20 conv     128       3 x 3/ 1     26 x  26 x 128 ->   26 x  26 x 128 0.199 BF
  21 conv     128       3 x 3/ 1     26 x  26 x 128 ->   26 x  26 x 128 0.199 BF
  22 route  21 20                                   ->   26 x  26 x 256
  23 conv     256       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x 256 0.089 BF
  24 route  18 23                                   ->   26 x  26 x 512
  25 max                2x 2/ 2     26 x  26 x 512 ->   13 x  13 x 512 0.000 BF
  26 conv     512       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x 512 0.797 BF
  27 conv     256       1 x 1/ 1     13 x  13 x 512 ->   13 x  13 x 256 0.044 BF
  28 conv     512       3 x 3/ 1     13 x  13 x 256 ->   13 x  13 x 512 0.399 BF
  29 conv      42       1 x 1/ 1     13 x  13 x 512 ->   13 x  13 x  42 0.007 BF
  30 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
  31 route  27                                      ->   13 x  13 x 256
  32 conv     128       1 x 1/ 1     13 x  13 x 256 ->   13 x  13 x 128 0.011 BF
  33 upsample                 2x     13 x  13 x 128 ->   26 x  26 x 128
  34 route  33 23                                   ->   26 x  26 x 384
  35 conv     256       3 x 3/ 1     26 x  26 x 384 ->   26 x  26 x 256 1.196 BF
  36 conv      42       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x  42 0.015 BF
  37 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 6.800
avg_outputs = 300731
 Allocate additional workspace_size = 12.46 MB
Loading weights from backup/yolov4-tiny_custom_last.weights...
 seen 64, trained: 448 K-images (7 Kilo-batches_64)
Done! Loaded 38 layers from weights-file
```

Figure A8 : Map calculation

Model testing

```python
#utility function
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline

  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)

  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  #plt.rcParams['figure.figsize'] = [10, 5]
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()
```

```
!darknet/darknet detector test custom_data/labelled_data.data darknet/cfg/yolov4-tiny_testing.cfg "backup/yolov4-tiny_custom_last.weights" all_note.jpg -ext_output
imShow('predictions.jpg')
```

```
  34 route  33 23                              ->   26 x  26 x 384
  35 conv    256       3 x 3/ 1     26 x  26 x 384 ->   26 x  26 x 256 1.196 BF
  36 conv     42       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x  42 0.015 BF
  37 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 6.800
avg_outputs = 300731
 Allocate additional workspace_size = 12.46 MB
Loading weights from backup/yolov4-tiny_custom_last.weights...
 seen 64, trained: 448 K-images (7 Kilo-batches_64)
Done! Loaded 38 layers from weights-file
 Detection layer: 30 - type = 28
 Detection layer: 37 - type = 28
all_note.jpg: Predicted in 15.693000 milli-seconds.
five_taka: 98%  (left_x:  396   top_y:  451   width:  672   height: 1298)
fifty_taka: 44% (left_x:  405   top_y: 1685   width:  650   height: 1080)
one_hundred_taka: 68%   (left_x: 1083   top_y:  496   width: 1402   height:  608)
thousand_taka: 50%      (left_x: 2457   top_y: 1139   width: 1536   height:  710)
two_hundred_taka: 99%   (left_x: 2550   top_y: 1893   width: 1440   height:  668)
fifty_taka: 100%        (left_x: 2598   top_y:  496   width: 1289   height:  677)
Unable to init server: Could not connect: Connection refused
```

Figure A9 : Model testing

```
(predictions:1444): Gtk-WARNING **: 09:53:25.504: cannot open display:
```

Figure A10 : Test result

Result Generate

```
[ ]  1 !darknet/darknet detector demo custom_data/labelled_data.data darknet/cfg/yolov4-tiny_testing.cfg "backup/yolov4-tiny_custom_last.weights" -dont_show 200.mp4 -i 0 -out_filename results_200_taka.mp4

two_hundred_taka: 100%

FPS:16.1        AVG_FPS:15.2

 cvWriteFrame
Objects:

two_hundred_taka: 100%

FPS:15.4        AVG_FPS:15.2

 cvWriteFrame
Objects:

two_hundred_taka: 100%

FPS:16.5        AVG_FPS:15.2

 cvWriteFrame
Objects:

two_hundred_taka: 100%

FPS:16.3        AVG_FPS:15.2

 cvWriteFrame
Objects:

two_hundred_taka: 100%
```

Figure A11 : Demo video creation