Deep Learning Based Ethiopian Car's License Plate Detection and Recognition

Anusuya Ramasamy, Joseph Wondwosen

Abstract: - Digital Image Processing is application of computer algorithms to process, manipulate and interpret images. As a field it is playing an increasingly important role in many aspects of people's daily life. Even though Image Processing has accomplished a great deal on its own, nowadays researches are being conducted in using it with Deep Learning (which is part of a broader family, Machine Learning) to achieve better performance in detecting and classifying objects in an image. Car's License Plate Recognition is one of the hottest research topics in the domain of Image Processing (Computer Vision). It is having wide range of applications since license number is the primary and mandatory identifier of motor vehicles. When it comes to license plates in Ethiopia, they have unique features like Amharic characters, differing dimensions and plate formats. Although there is a research conducted on ELPR, it was attempted using the conventional image processing techniques but never with deep learning. In this proposed research an attempt is going to be made in tackling the problem of ELPR with deep learning and image processing. Tensorflow is going to be used in building the deep learning model and all the image processing is going to be done with OpenCV-Python. So, at the end of this research a deep learning model that recognizes Ethiopian license plates with better accuracy is going to be built.

Keywords: Amharic Characters, Deep Learning, Image Processing, License Plate, OpenCV- Python, Tensorflow

I. INTRODUCTION

In recent years, the number of privately-owned cars has increased considerably and this has, in turn, exacerbated the traffic management burden. The resultant congestion has caused extreme problems such as traffic accidents or public space vulnerability to crime or terrorist attacks [1].

As a result, various Intelligent Transportation Systems (ITSs) emerge as viable solutions to those problems. ITSs apply technologies of information processing and communication on transport infrastructures to improve the transportation outcome. One of these systems is License Plate Recognition (LPR) which is the most popular and important element of ITS [2].

The main job of LPR systems is to detect the license plate from rest of objects found in an image and recognize the Alphanumeric characters which in case of Ethiopian license plates are Amharic characters, Digits (ranging from 0-9) and in most cases English characters also [2].

Revised Manuscript Received on March 30, 2020.

***** Correspondence Author

Dr. Anusuya Ramasamy*, Assistant Professor, Faculty of Computing and Software Engineering, AMIT, Arbaminch University, Ethiopia.anusuccess@yahoo.com

Mr. Joseph Wondwosen[2], [2] Research Scholar, Faculty of Computing and Software Engineering, AMIT, Arbaminch University, Ethiopia. joshu0916@gmail.com

Till date there is only one research that that has been conducted specifically on Ethiopian license plates. It was attempted using conventional Image Processing algorithms and was able to achieve an overall accuracy of 63.1%.

In this research an effort is going to be made in detecting and recognizing Ethiopian license plates using Deep Learning. The basic advantage in using deep learning is that unlike the traditional template machining algorithms of image processing, feature extraction is done through directly learning from images, text or sound [3] [4] [5]. As, a result deep learning models can achieve state of art accuracy, sometimes exceeding human-level performance [6]. LPR systems basically contain three phases or stages: Plate Detection, Character Segmentation and Character Recognition. Plate Detection is categorized under one of the broader typical tasks of computer vision that is Object Detection [7]. Object Detection is scanning or going through an image in search of a specific object and there are so many algorithms that has been used till date in fulfillment of this task. Before emergence of deep learning different mathematical models which are based on some prior knowledge like Hough Transform, Frame Difference, Background Subtraction, Optical Flow, Sliding Window and Deformable Part methods were used and are still used to some extent, but currently deep learning-based algorithms are showing a state-of-art performance when it comes to object detection [8]. In this research an effort is going to be made in trying to detect the license plate from an image using deep learning approach. The second phase is Character Segmentation which is performed on the detected or extracted plate image. This phase aims at separating each character found on a detected plate and feeding them individually to the next phase, to be recognized. There are many image processing techniques and libraries which are used for image segmentation and this research will use OpenCV-Python library for its image preprocessing and segmentation tasks. The third and final phase is recognition where each character that has been segmented earlier gets recognized or classified. There are many traditional image processing techniques like Histograms of Oriented Gradients which are used to represent an image to be classified but they need domain expert's knowledge and guidance for feature extraction. There is also a deep learning approach which automatically learns and extracts those features by itself which in turn has a dramatic impact on the performance of classification task [9]. Overall in building the deep learning models a library called Tensorflow is going to be used and for all other image rocessing tasks the research will use OpenCV-Python

II. STATEMENT OF PROBLEM

It is known that number of motor vehicles on the main roads of Ethiopian cities are increasing considerably every year which as a result causes high traffic congestion problem. The congestion problem in turn leads to many other problems like loss of life from traffic accidents, vehicle thefts and other security related problems. Even though this problem is being addressed through Intelligent Transportation Systems (ITSs) in most of the developed countries through research, in Ethiopia there has only been one research and it was attempted using a conventional OCR based Image Processing. In case of License Plate Recognition, image of a car whose plate is to be processed may have an angle where the plate is partially visible, low resolution quality, varying distance from the camera and low lighting conditions which makes is very hard (challenging) to process and recognize. In this section different papers related to this research are going to be reviewed. The review is going to mainly focus on a problem or research gap that the reviewed paper tries to solve and methodology it uses.

Although this is a popular research area and it has been attempted using so many techniques including traditional image processing, machine learning and deep learning, since, license plates of different countries have different features like size of the plates, characters contained in the plate (which in this research's case are Amharic letters, English letters and Numbers ranging from 0-9) and methodology used, it is still worth investigating. In case of Ethiopian license plates there is only one research till date and it has been attempted with traditional image processing. But other technologically developed countries like China, America, India, UAE and many more have exploited and conducted many researches regarding the state-of-art deep learning models in recognizing their own language's character features which are included in their country's respective license plates.

III. RELATED WORKS

A. Researches on Ethiopian License Plates

This paper was conducted on recognition of Ethiopian license plates. It used Gabor Filtering, Morphological Closing Operation and Connected Component Analysis for plate region detection. Gabor filter is performed on gray scale image and then its response is binarized to perform morphological closing. Finally, connected component labeling is performed to find connected objects or components in a resultant image. In segmentation phase first it applies orientation adjustment and size normalization by performing Canny edge detection and Hough transform. Second it performs segmentation with connected component labeling, finally, numbers and other characters are separated based on their location and size (i.e. width and height). Once characters of the plate are segmented correlation-based template matching is performed for recognition.

B. Researches on Non-Ethiopian License Plates

This paper basically, has three major phases: plate detection, character segmentation and character recognition where 2 separate CNN models are used for detection and recognition. The detection phase has both image preprocessing and CNN classification (detection) steps. Here the preprocessing phase consists of morphology filtering to contrast maximization, gaussian blur filter to remove noise, adaptive threshold to eliminate unimportant regions in the image, finding all contours to locate a curve that joins all continuous points having the same intensity, geometric filtering to improve the precision of LP detection, CNN detection and drawing boundary boxes around plate region with minimum threshold value of 0.7. Segmentation phase consists of gray scaling, canny edge detection, extraction of contours, geometric filtering and boundary boxes of characters segmented. Finally, in last phase which is character recognition the second CNN model is used.[12]

This paper is about classification of images as license plate or non-license plate by using CNN. The network used was constructed of 7 layers, where the first convolutional and sub-sampling layers have 4 feature maps and the next once after that have 11. The third convolutional layer has 120 feature maps it is connected to the fully-connected layers which has 84 units. Finally, it has an output layer which classifies the image as either a license plate or non-license plate.[13]. This paper was proposed efficient hierarchical methodology for license plate recognition system by first detecting vehicles and then retrieving license plates from detected vehicles to reduce false positives. Finally, CNN based LPR is used in recognition of plates characters. It used YOLO V2 for vehicle detection which has 19 convolutional layers and 5 max pooling layers. For license plate detection it used SVM and for character segmentation it performed a number of image processing techniques like: gray scaling, binarization, horizontal and vertical projection. For character recognition it used a CNN model with 2 convolutional layers, two ma pooling, two fully connected layers and one output layer. [14]

This paper is pure image processing based and it has 4 basic steps: preprocessing, localization, segmentation and recognition. In preprocessing phase, it performs gray scaling and median filtering to get rid of the noise while conserving sharpness of the image. In localization phase, region of the plate is detected from rest of objects in an image. [15].This paper aims at demonstrating the capability of CNN in recognizing vehicle's states (or regions) from a number plate. The researchers only considered 4 classes for simplicity sake. For each class or state the dataset contained 200 images where each image had some kind of distortion, tilt and illumination at different angles. The results achieved are more than 95% in average. [16]

This paper is proposed an end-to-end deep-learning based ALPR system for Brazilian license plates. The system presents two considerably fast and small YOLO based networks operating in cascade mode. Since searching for a relatively very small object such as a license plate from a higher resolution image demands too much of computing resource, the paper first performs frontal view extraction where it extracts the front of a car which in turn contains the license plate.[17]. CNN has some advantages for feature extraction on the shapes of the image such as parameters of spatial filters. However, as presented on result discussion the RGB and gray scale color channels of the coffee bean have shown significant impact on classification accuracy of the model. The result was showed that the classification accuracy of the optimal model was 98.38% and 91.43% for RGB color image and gray scale images respectively. The test result showed that using test datasets the three categories of coffee grades, grade 1 grade2 and grade 3 was identified classified 99.51%, 97.56%, and 98.04%, respectively, and the overall performance was 98.38%. [10]

IV.RESEARCH METHODOLOGY

This chapter of the document discusses about the proposed design that this research employs. The first section will discuss in detail about different properties of Ethiopian license plates. The second section will discuss about proposed design for each phase individually and finally the third section will discuss about how each module from each corresponding phase is going to communicate with each other and work as a single system.

A. Features of Ethiopian license plate

Ethiopian license plates can basically be categorized based on plate code and its region. Plate codes are represented both in numeric and character format where the numeric code ranges from 1 to 5 and the character codes which can either be Amharic or English have around 8 categories. Considering regional classification, Ethiopia has 9 national regions (i.e. Tigray, Afar, Amhara, Oromia, Somali, Benishangul-Gumuz, Southern Nations Nationalities and People Region (SNNPR), Gambella and Harari) and 2 administrative states (i.e. Addis Ababa city administration and Dire Dawa city council) where each can print their own plates of code: “1-5”, “የ ዕ ለ ት”, “ተ ላ ላ ፊ” and “ል ዩ ”. So, in order to identify a given plate the LPR system must be able to get the its code and regional information since a same plate number can be given to different vehicles in different regions and plate codes. Morphologically speaking Ethiopian license plates have two formats that are single row and double row plates.

Table I: Ethiopian Car's License Plate Color Properties Based on There Code

EN/AM Code	**Foreground Color**
CD/ኮዲ	Black
AU/አህ	Light Green
AO/ዕድ	Orange
UN/የተመ	Light Blue
-/ፖሊስ	Black

EN/AM	**Foreground**
-/የዕለት	Red
-/ተላላፊ	Light Blue
-/ልዩ ተ	Red

B. Proposed Design

In order to build a deep learning model for license plate recognition which contains both image detection and classification tasks, the research has to go through many stages which are shown in an overall design diagram below:

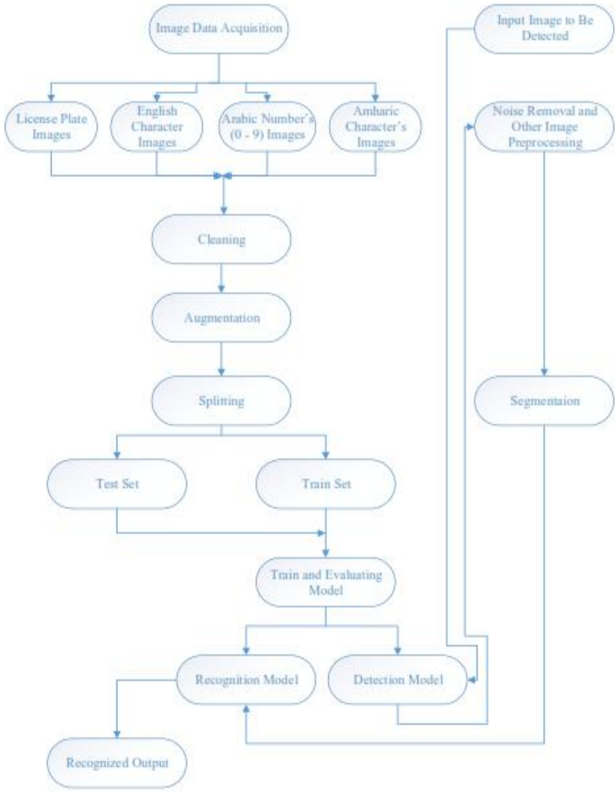

Fig .1. Overall System's Design Diagram

C. Image Data Acquisition

In this part of the research, images of the different cars with different variations of license plates have been captured using a digital camera with a resolution of 13 MP. Addis Ababa were chosen as a place of data collection because it is a capital city of Ethiopia and so, different cars from different regions of the country having different regional code can be found there. The images were taken from different distance, camera angles and illumination in order improve model’s detection and recognition accuracy under different conditions and circumstances. The images were also captured while the car is moving with slow speed, medium speed and high speed. The images needed to train the recognition model (English letters, Numbers and Amharic letters) were taken from cropped from the collected license plate images.

D. Cleaning

In this part of the research before the images were used in training there was image pyramiding step. Pyramiding is a representation where the image has to go through series of smoothing and subsampling. There are two types of pyramiding, Gaussian and Laplacian pyramids. In this research in order to make the model training process faster a down pyramiding with Gaussian filter was used. Here 5 pixels from the original image with higher resolution are going to contribute or gets reduced to one pixel in a resulting lower resolution image [20]. So, if the original image were M x N it will be M/2 x N/2. All the images used in detection model were reduced to height less than 1600 pixels and a width less than 1200 pixels.

E. Augmentation

Image data augmentation is a crucial stage while developing deep learning models. It enables expanding the amount of training image data in a given training dataset by creating modified versions of images in a dataset. Even though there are different augmentation techniques out there, in this research a Random Horizontal flip is used which is basically reversing the columns of pixels. Other augmentation styles like vertical or horizontal shift weren't used because in case of shifting some part of the plate might be cutoff and that makes recognition impossible.

F. Splitting

Splitting is a process of separating the whole dataset in to Train and Test set, where a train set will be used by a model to learn different categories or classes by making predictions on input image and making corrections when the prediction that has been made is wrong. Test dataset is used to evaluate the performance of the model once it has been trained. Even though there is no rule in choosing the proportion of what train and test set size to use, after reading many blogs and books

80 to 20 percent ratio (80% for train and 20% for test) has been used splitting the entire dataset. So, for the detection model out of 1100 collected car images 880 were used for train set and the rest 220 were used for test set. And for classification model out of 4240 total cropped character images 3392 were used for train set and the rest 848 were used for test set.

G. License Plate Detection

The license plate detection part was developed using deep learning approach but while applying this approach for object detection there is one tradeoff that must be considered depending on the problem and that is Speed/Accuracy. So, considering license plate detection, accuracy is more important feature to consider than speed because missing even one character from the detected plate means not identifying the vehicle as a whole. So, in this research a better model (when it comes to accuracy) called Faster R-CNN is going to be used [18]. This model is mainly composed of two modules. The first one is a fully convolutional network that proposes regions letting it have two primary benefits, like being fast and able to accept images of varying resolution having any width and height. The second one is the Fast R-CNN detector which uses the proposed regions from the first module.

H. Faster R-CNN Detection

So, for feature extraction although the original Faster R-CNN paper has used VGG and ZF as base networks, in the proposed system deeper and more accurate ResNet will be used [4]. ResNet is a much deeper network than its previous architectures like VGG16 and VGG19 architectures. It uses residual model to train CNNs which have over 1000 layers on CIFAR-10 [19]. In order to reduce the volume size ResNet uses only two pooling layers, the first one is Max Pooling which is used at the beginning to reduce the special dimensions and the second one is Average Pooling at the end. Unlike the common CNNs ResNet adds the original input to the outputs of convolution, ReLU and BN layers, this operation is called Identity Mapping and the reason for using the term Residual. In this research ResNet 101 has been used as base network for feature extraction so it has 101 layers.

I. Plate Extraction

Once the plate has been detected or located, the detection model provides a list of coordinates of possible plate region with their corresponding score results. So, the extraction phase will find the highest scoring bounding box crop it out. The result of the extraction phase (i.e. cropped plate image) will be an input to the segmentation phase

J. Character Segmentation

The segmentation of characters on the detected license plate is done with a combination of different digital image processing techniques. OpenCV or Open Computer Vision with python has been used to do each image processing task including all necessary preprocessing or cleaning.

1. Orientation Adjustment

Before segmenting the characters on the plate, the orientation of the plate should be in a correct format (i.e. horizontally aligned). So, in order to do that first angle of the longest line, which is either the upper or the lower border of the plate needs to be found. First the input image is gray scaled and then the all the edges are found using Canny edge detector. Canny edge detector is a multi-stage edge detector which comprises 4 stages. On its initial phase, it does Noise Reduction with 5 x 5 Gaussian filter and then it finds the Intensity Gradient of the image by filtering it in both horizontal (Gx) and vertical (Gy) direction. The gradient will always be perpendicular to the edges. It uses the below equations:

Equation 4-1: Equation of Canny Edge Detector

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$$
$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Once the gradients magnitude and direction has been found the next step is Non-Maximum Suppression where pixels that doesn't constitute an edge are removed. Finally, with Hysteresis Thresholding by using 2 threshold values (min and max) it decides whether an edge is really edge or not. If the edge's intensity value is more than max then it is sure-edge else if it is less than min it is not. If intensity value falls in between max and min value the algorithm checks its connectivity. So, if it is connected to sure-edge then it is an edge else it is not. Once all the edges are found using Canny next step is to find edges that constitute a line using HoughLines method that takes four parameters. First parameter is a binary image that has been found using canny, second and third parameters are *rho and theta* values which are measured in pixels and radians respectively. The fourth argument is a threshold value that specifies the least vote it should get in order to be considered as a line. Finally, by using *theta* value (angle) of the longest line, the transformation matrix is found using the below equation

4-2: Equation for Transformation Matrix in HoughLines

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot center.x - \beta\cdot center.y \\ -\beta & \alpha & \beta\cdot center.x + (1-\alpha)\cdot center.y \end{bmatrix}$$

where:

$$\alpha = scale\cdot\cos\theta,$$
$$\beta = scale\cdot\sin\theta$$

The transformation matrix is given to warpAffine method which takes 2 x 3 transformation matrix and returns the transformed image based on the matrix given.

2. Border Elimination

Here the upper and lower borders of the horizontally oriented plate image are eliminated. The first step is to determine whether the image has one or two rows. So, the aspect ratio of the image is taken into consideration. Once the number of rows is known HoughLines is used to get either the upper of the lower border (since the longest line is always one of the two) of a plate image. If the plate image has one row then the height of the largest contour is going to be either added or subtracted from the y value of the detected line. If the detected line is bottom of the plate then height of the largest contour is going to be subtracted else it gets added. So, the resulting area is going to be taken as ROI (Region of Interest) and gets cropped out.

Fig.2. Border Elimination Sample

3. Segmentation

Here all the characters on the plate are going to be segmented into individual letters and numbers. So, the output image from border elimination is going to be binarized using OTSU method which works on bimodal images. Bimodal image is an image whose histogram has two peaks. So, in OTSU unlike global binarization methods where an arbitrary value is taken as a threshold, it takes an approximate value that is in a middle of the two peaks of a bimodal image. OTSU algorithm tries to find a threshold value (t) which minimizes the *weighted within-class variance* given by relation equation.

4-3: Equation through which OTSU Finds Threshold Value (t)

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

where

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_1(t) = \sum_{i=t+1}^{I} P(i)$$
$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$
$$\sigma_1^2(t) = \sum_{i=1}^{t} [i-\mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i-\mu_1(t)]^2 \frac{P(i)}{q_2(t)}$$

Once an image has been binarized the next step is to find all the contours. Contour is simply a curve that joins all continuous pixels that have same intensity level. It takes three arguments where the first is source image, second is contour retrieval method and third is contour approximation method. Approximation method determines whether the contour holds all the coordinates of the boundary or not. In this research a method called CHAIN_APPROX_SIMPLE is used which removes redundant points and compresses the contours in order to save memory.

Although there are different kinds of contour retrieval modes, this research used RETER_EXTERNAL because of the unique properties that Ethiopian license plates have. For some specific characters like number codes, rather than taking circle shape and the number code inside it separately RETER_EXTERNAL takes the circle and the number code all at once. Finally, noises (i.e. non-characters) are going to be filtered based on their contour area. So, any contour that does not have properties of a plate character is going to be removed. So, valid contours are going to be cropped and saved as separate images.

Fig.3. Plate Character Segmentation Sample

4. Character Recognition

In this part a convolutional neural network (CNN) has been trained to classify or recognize the characters on the plate. Plate characters or images have 3 channels (RGB) with both height and width having a size of 28. ReLU (Rectified Linier Unit) has been used as an activation function except for the outer layer of the network where SoftMax has been used since it is preferable choice when it comes to a deep learning model with more than 2 classes [21]. ReLU unlike the old Sigmoid and Tanh functions it is not saturable meaning the gradients doesn't get killed when neurons saturate (i.e. It doesn't have a vanishing gradient problem). It is also extremely computationally efficient and sparsely activated meaning there is a strong likelihood for any given unit not to activate at all since it is zero for all negative inputs. And it shows better performance being applied on different application areas [4]. Visually: As a final layer an activation function called SoftMax which mostly used in multiclass classification problems [22]. It extends the idea of logistic regression which produces a decimal between 0 and 1 into a multi-class problem by assigning probabilities for each class in that problem. Those decimal probabilities must finally add up to 1.0. SoftMax should be used with mutually exclusive classes because it predicts only one class at a time. Below is its equation:

Equation 4-4: Equation for SoftMax Activation Function

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K}\exp\left(s_j(\mathbf{x})\right)}$$

Where K is the number of classes, s(x) is a vector containing the scores of each class for instance x, $\sigma(\mathbf{s}(\mathbf{x}))_k$ is the estimated probability that the instance x belongs to class k given the scores of each class for that instance.

Retrieval Number: F9857038620/2020©BEIESP
DOI: 10.35940/ijrte.F9857.038620

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The number of samples preprocessed before updating the model (batch size) is 1 and an iterative algorithm, SGD (Stochastic Gradient Descen) has been used as an optimizer while training. SGD is a modification to standard Gradient Decent algorithm. After computing the gradient, rather than updating the weight matrix on the whole training data it only updates on small batches or samples. This make the algorithm much faster since the data or number of samples being manipulated at each iteration is really small. The overall network looks like, As it can be seen in a table the network has 2 convolutional layers, 2 pooling layers, 3 ReLU activations, 2 fully connected layers and finally a SoftMax activation. Max pooling has been used for all pooling layers. Pooling basically reduces the special size of the input image which in turn reduces the number of parameters and computation in the network. Although a pool size more than 2 x 2 can be used for larger input images, for smaller sized ones like those used in this research 2 x 2 is an appropriate choice.

Table II: Character Recognition Model's Architecture

Layer Type	**Output size**	**Filter size /**
Input Image	28 x 28 x 3	
Conv	28 x 28 x 20	5 x 5, k = 20
ACT	28 x 28 x 20	
POOL	14 x 14 x 20	2 x 2
Conv	14 x 14 x 50	5 x 5, k = 50
ACT	14 x 14 x 50	
POOL	7 x 7 x 50	2 x 2
FC	500	
ACT	500	
FC	16	
SoftMax	16	

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter of the research the results of the proposed system are going to be evaluated and its outcome is going to be discussed. So, the first part is going to be about object detection, the second part is going to be about character segmentation and finally the third part is going to be about character segmentation.

A. Object Detection

The object detection model takes an image which contain different objects including the car and detect or localize only the license plate. the model was trained with 1100 different images that include both the car and the license plate. The license plates images in the training dataset are composed of different plates with varying regional code and distance from the camera so that the model may be able to generalize more.

Fig .4. Sample Detection Results

The whole dataset is divided into training and testing set with a ratio of 80 to 20 respectively. So, the training set contains 880 images and the testing set contains 220 images. The original images were taken with a 13 MP camera that has an image resolution of 3120 x 4160 which is too big for training. So, image pyramiding was used in order to reduce the resolution of the image. After the pyramiding a resolution of 3120 x 4160 image was reduced to 780 x 1040.The model was trained with 50000 steps and a batch size of 1. It has a learning rate of 0.0001 which were scheduled to decrease to 0.00001 after 90000 steps and to 0.000001 after 120000 steps, but since the loss stopped to drop down after the 40000's step training process was terminated after going 50000 steps so the learning rate remained at 0.0001.So, after training the following results:

```
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.991
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.908
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.760
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.762
```

Fig.5. ResNet Trained Detection Model's Evaluation Result in AP an AR

The box classifiers classification and localization loss are 0.018383 and 0.012104 respectively. Among the nine images which were randomly chosen during model evaluation the license plates were detected successfully in all cases. The same dataset was trained with Inception v2 model having a learning rate of 0.0002 and a batch size of 1. And it had lower MAP @ 0.75 IOU and AR @ 0.50 IOU that is 0.849132 and 0.732857 respectively, which results in the detected plate having some of its part being cutoff. Below are its results:

```
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.996
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.849
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.733
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.733
```

Fig.6.Inception V2 Trained Detection Model's Evaluation Result in AP an AR

So, the proposed model not only showed a better detection accuracy than a model by [2] which was developed with conventional image processing techniques having a detection accuracy of 88.9, it also performed better than a deep learning model trained with Inception feature extractor. Below is comparison of the model with some related works:

Table III. Comparison of Ethiopian License Plate's Detection Model With Some Related Works

Related Works	Accuracy
YOLO based detection [34]	94.23 %
Fast-YOLO [37]	95.07 %
Plain CNN [32]	96 %
Proposed System	99.1%

As can be seen from the table above the proposed model achieved a better accuracy comparatively

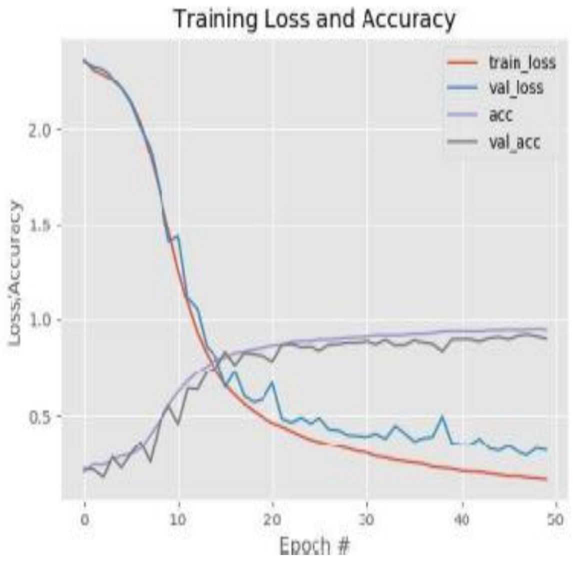

Fig .7. CR Model's First Experimental Evaluation Results

Table IV: Justification Summary Table for the Model

Model Name	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Average Precision On Each Class
Model-1	0.1697	0.9485	0.3204	0.9038	0.91
Model-2	0.0017	0.999	0.2281	0.9550	0.96
Model-3	0.000029	0.999	0.1488	0.98	0.98
Model-4	0.2255	0.9440	0.3190	0.9150	0.92
Model-5	0.00019	0.999	0.1481	0.9652	0.97

VI CONCLUSION AND FUTURE WORK

A. Conclusion

This research studied about a deep learning-based approach for recognition of Ethiopian car license plates. This research is important because Ethiopian license plates has their own unique features and this problem has only been approached with conventional Image processing method. This study has three main parts: License Plate Detection, Character Segmentation and Character Recognition. Once the license plate is detected its characters are segmented using image processing methods. So, the segmentation module takes an image: preprocess it, performs orientation adjustment, removes the borders and finally segment each alpha numeric character. It was built with OpenCV- python. The segmentation model achieved 86.66%. The segmented characters are going to be given to the classification or recognition model which was developed using Convolutional Neural Network. The CNN model classifies each character image to its corresponding class. It achieved a pretty good classification accuracy in both training and validation set which is 99.9% and 98% respectively.

REFERENCES

1. X. Lele, . A. Tasweer and J. Liyanwen , "A New CNN-Based Method for Multi-Directional Car License Plate Detection," IEEE, p. 11, 2018.
2. S. Nigussie and A. Yaregal , "Automatic Recognition of Ethiopian License Plates," IEEE, p. 5, 2015.
3. Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," Nature, vol. 521, p. 9, 2015.
4. A. Rosebrock, Deep Learning for Computer Vision with Python, PyImageSearch, 2017.
5. I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.
6. Mathworks, "Mathworks," [Online]. Available: www.mathworks.com/discovery/deep- learning.html. [Accessed 18 12 2018].
7. "Wikipedia," 9 May 2019. [Online]. Available: https://en.m.wikipedia.org/wiki/Computer_Vision. [Accessed 9 May 2019].
8. C. Tang, Y. Feng, Y. Xing, C. Zheng and Y. Zhou, "The Object Detection Based on Deep Learning," IEEE, p. 6, 2017.
9. N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional Neural Networks for Image Classification,"IEEE, p. 6, 2018.
10. J. R. Arunkumar, Tagele berihun Mengist," Developing Ethiopian Yirgacheffe Coffee Grading Model using a Deep Learning Classifier" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-4, February 2020.
11. "Omnitec," 12 February 2018. [Online]. Available: www.omnitecgroup.com/blog/advantages-of-automatic-license-plate-recognition-system-6545. [Accessed 1 May 2019].
12. Z. Selmi, M. B. Halima and A. M. Adel , "Deep Learning System for Automatic License Plate Detection and Recognition," IEEE, p. 7, 2017.
13. Z. Zhihong, Y. Shaopu and M. Xinna , "Chinese License Plate Recognition Using a Convolutional Neural Network," IEEE, p. 4, 2008.
14. C.-H. Lin, Y.-S. Lin and W.-C. Liu, "An Efficient License Plate Recognition System Using Convolution Neural Networks," IEEE, p. 4, 2018.
15. P. Prabhakar, A. P and R. S. R, "Automatic Vehicle Number Plate Detection and Recognition," IEEE,p. 7, 2014.
16. M. Mondal, P. Mondal, N. Saha and P. Chattopadhyay, "Automatic Number Plate Recognition Using CNN Based Self Synthesized Feature Learning," IEEE, p. 4, 2017.
17. S. Montazzolli and C. Jung, "Real-Time Brazilian License Plate Detection and Recognition UsingDeep Convolutional Neural Networks," IEEE, p. 8, 2017.
18. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE, p. 14, 2016.

Retrieval Number: F9857038620/2020©BEIESP
DOI: 10.35940/ijrte.F9857.038620

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

19. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," IEEE, p. 9,2015.
20. A. Mordvintsev and A. K, "OpenCV-Python Tutorials Documentation," 5 Nov 2017. [Online]. Available:https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials.html. [Accessed 1 May 2019].
21. A. Kızrak, "Towards Data Science," 9 May 2019. [Online]. Available:ttps://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a. [Accessed 15 Oct 2019].
22. "Google Developers," [Online]. Available: https://developers.google.com/machine-earning/crashcourse/multi-class-neural-networks/softmax. [Accessed 20 Dec 2019].

AUTHORS PROFILE

Dr. Anusuya Ramasamy has many achievements in engineering and Technology in India and Abroad. She has B.E. M.E, Ph.D. in the field Computer Science and Engineering. She has served more than 12 years in Academic of research/Articles/journals/funded projects. Currently, she is working more than 6 years as an Assistant professor in faculty of Computing and Software Engineering, Institute of Technology, Arbaminch University, under the MOEFDRE, UNDP projects in Ethiopia. She published more than ten International and National journals like Scopus and UGC approved journal. He also obtained one patent for his work

Mr. Mr. Joseph Wondwosen MSc Scholar at Arba Minch University, He has BSc degree in Software Engineering from Adama science and Technology University,2014. His Research Interest is Artificial Intelligence, Data Science and Image Processing., he has excellent contribution in higher technical education.