WHITE PAPER
www.visteon.com

# Multilingual String Verification for Automotive Instrument Cluster Using Artificial Intelligence

# Multilingual string verification for automotive instrument cluster Using Artificial Intelligence

Deepan Raj M [a], Prabu A [b]

Software Engineer at Visteon Technical & Services Centre, Chennai [a]

Technical Profession at Visteon Technical & Services Centre, Chennai [b]

## Abstract:

*Validation of HMI contents in Driver information system, In-Vehicle Infotainment, and Centre media Display is a big challenge as it prone human manual testing errors. Automating the text verification for multi languages and horizontal/vertical scrolling text is difficult challenge as the intelligence needed to achieve is high. To mitigate this effect vision based machine/deep learning algorithms will be implemented in such a way that three important applications have been created which serves to be the mother code for all the features that involves text as it main component of automated testing. They are*

1. *Multilingual text verification at various illuminations, grade out conditions and various background gradients using Opencv and Tesseract-OCR.*
2. *Horizontal & Vertical scrolling text using image key point stitching.*
3. *Automatic region of interest generation for text regions in the message center display using inception v3 deep neural network architecture.*

*With the power of artificial intelligence in validating text makes it easier to automate with greater accuracy and can also be implemented in other areas like User Setting Menu traversal verification, Clock verification and much can be automated with less time and higher accuracy.*

Keywords: Computer Vision, Machine Learning, Deep learning, Neural Networks, OCR, automating validation.

**1. INTRODUCTION:**

In recent years there has been a drastic improvement in the field of automobile, all the analog and semi digital clusters are being changed to full digital cluster, and lot of driver assisting feature are being added as the days goes on. As the functionality increases the Driver information and Infotainment system will be loaded with lot of information's and since automobile goes multinational the information's will be provided in foreign languages to. For an example in Hyundai Ionic instrument cluster has 23 languages including English. Due to this validation of strings for all the languages have become time consuming and stressful task as the human error creeps in. The validator will not have knowledge of other foreign languages and it will again lead to error and time consumption. In some case the information will be scrolling as the display is short to show the entire string which makes the validation even more challenging. In order to mitigate this effect automation using artificial intelligence comes into play, by using Neural Networks power all forms of texts can be validated in touch of a button. There are tools are which is used like NI-OCR [1], Abby [2] etc for Optical character recognition, but none has the neural network enhancement which gives Tesseract-ocr [3] tool developed by google a greater edge, thus by using googles open source api Tesseract-Ocr and image processing techniques great accuracy is met at automation of string validation. The Tesseract is being utilized at multiple verticals as we have combined CNN and other machine learning techniques to capture other forms of texts like Horizontal and vertical scrolling text, and segmentation of text from the image (without manually cropping the portions of text manually) and much more.)

## 2. SYSTEM REQUIREMENTS:

For this application we have utilized the power of python libraries and Tesseract OCR to achieve end to end automation in various forms for text verification.

### 2.1. Tesseract-OCR:

Hewlett-Packard was the founder of Tesseract between 1985 and 1994 and this tool was made publically available in the year 2005, and from 2006 Google observed the tool since then google is actively maintaining it, and recently Tesseract has release an alpha version 4.0.0 where it actually utilizes the power of Recurrent Neural Network/ Long Short Term Memory. Because of the inclusion of Neural networks the system actually started frame words that gives meaning. And with this the error rate for all the languages is limited under 10 %. But Tesseract needs perfect high quality image without noise, since the images gathered are from real time noise will be high and the quality will be par. By using image processing techniques the noise and image quality can be somehow increased in such a way that Tesseract gets a neat image for extracting text.

### 2.2. Open CV:

Image processing techniques are done using this python library [4]. This is an open source and easy to use library where lot of matrix computation are handled internally making it very simple and efficient to code. Using open CV we will eliminating noise, enhancing the graded out text, stitching of images for scrolling text verification and feature extractions for image can be done using this library.

### 2.3. Tensorflow:

Tensorflow [5] is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. Tensorflow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

By using Keras as front-end and by keeping Tensorflow as backend coding complexity can be increased with minimum scarification of speed which is not a concern as the processing time is already in the limits. For segmenting texts (to find the portions of text in a given image) we need a powerful feature extraction technique and that is achieved using CNN by coding through keras-tensorflow combination.

### 2.4. SciKit Learn:

SciKit Learn [6] is another open source python library which is extensively used for training a machine learning model, the feature extracted data and labels will be trained using this framework, and it has almost all the important machine learning models. Using this we can train model (with the feature vectors and labels which is being extracted using Opencv and Tensorflow) with different classifier to know which classification model suits the best in terms of speed and accuracy.

### 2.5 Provetech:

In order Execute test procedures to the Instrument cluster and infotainment system, CAN signals and other hardware emulation can be commanded using this tool, and all the execution can be automated and the provetech [7] will call python for text verification and further result will be generated in provetech by getting feedback of the result from python.

With this combination any forms of text verification can be done, and it is not limited to text verification and can be extended to verify other features to namely gauges, odometer, telltales etc.

## 3. METHODOLOGY:

The paper explains about three techniques namely

1. Multilingual text verification at various illuminations, grade out conditions and various background gradients using Opencv and Tesseract-OCR.

2. Multi Lingual Horizontal & Vertical scrolling text using image key point stitching.
3. Automatic region of interest generation for text regions in the warning pages using deep neural network architecture.

### 3.1 Multilingual Text verification (OCR):

OCR[8] is nothing but Optical Character Recognition (i.e.) it can recognize character which is actually in the format of image or pdf. The OCR will achieve great accuracy if the characters in the words are well organized like shown in the below fig. 3.1

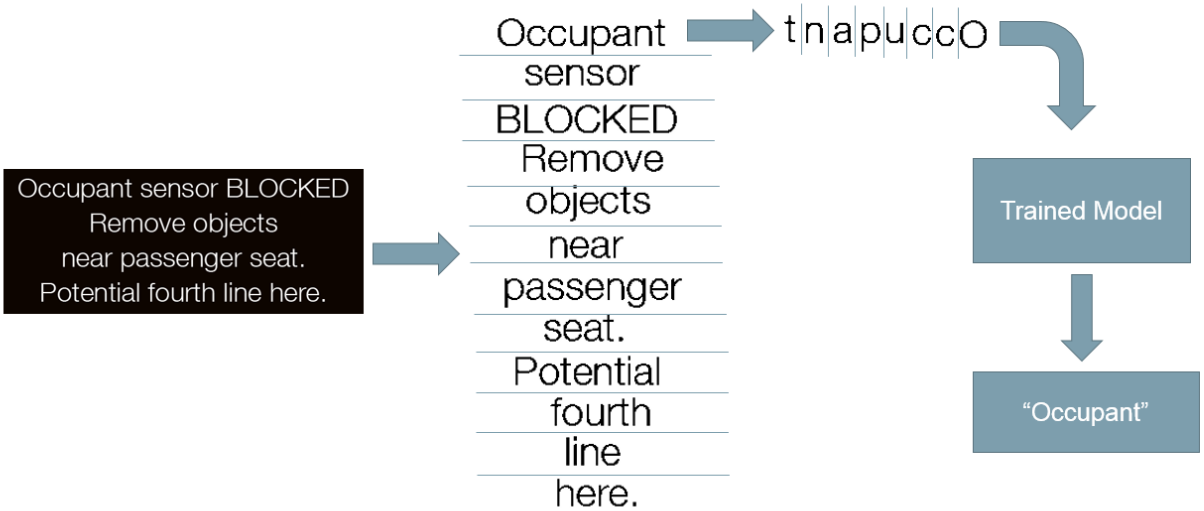


Fig.3.1 Normal OCR working

The biggest problem occurs in the character chopping mainly due to some properties like Kerning, Connected Character and interspace in single word as shown in fig. 3.2.

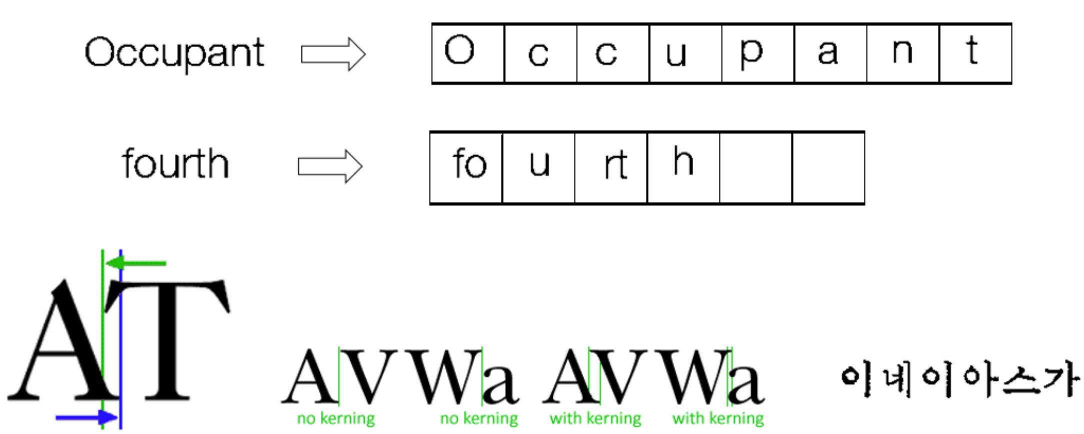


Fig3.2. problems in character chopping – connect character, kerning, interspace in single word

Thus by using the power of RNN/LSTM the previous learnt memory will be helpful in chopping the characters precisely to predict the words.

Long short-term memory (LSTM) [9] is a recurrent neural network (RNN) architecture that remembers values over arbitrary intervals. Stored values are not modified as learning proceeds. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events.

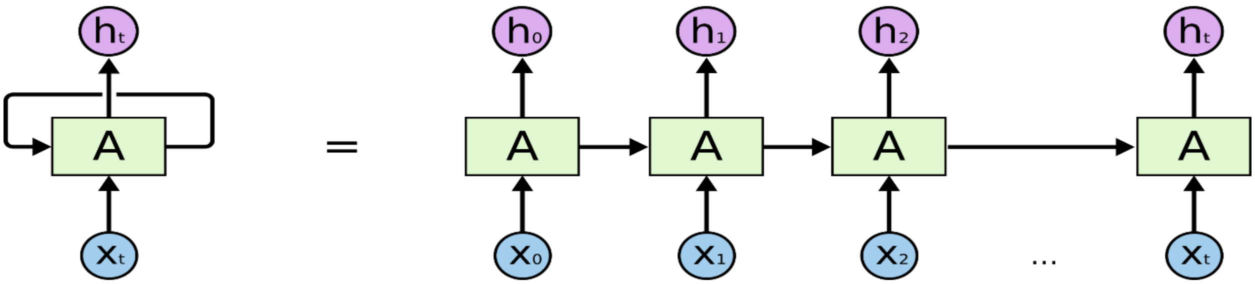


Fig 3.3 LSTM

In fig 3.3, Where Iteration of Model “A” with i/p “x” and o/p “h” until max probability achieved.

With the power of LSTM Tesseract has leaped the market with error less than 10 % for all the languages. Using this ideology an application that involves processing of images and applying it to Tesseract done using python as shown in the fig. 3.4.

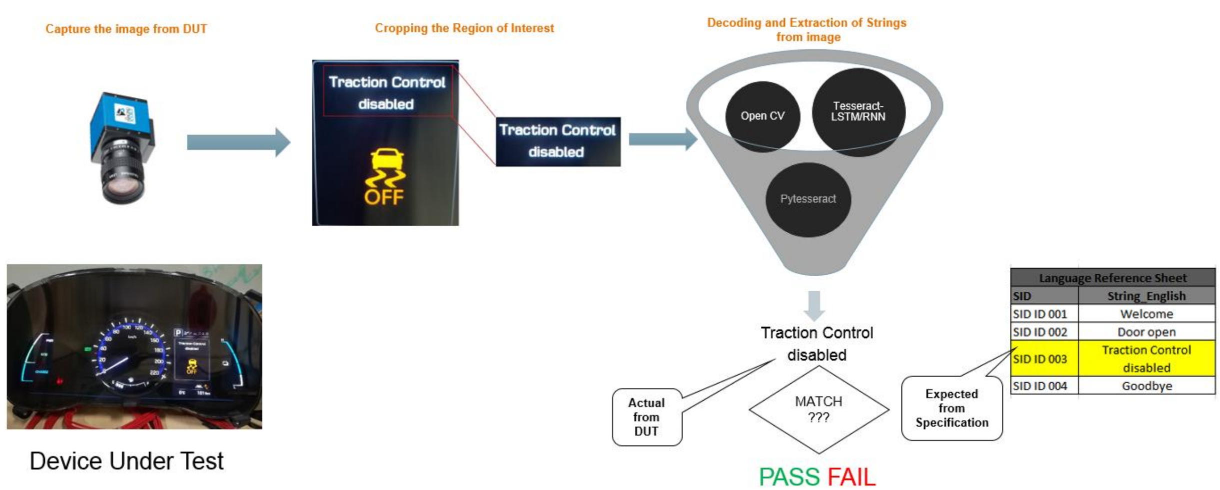


Fig 3.4 Automated Multi Lingual text verification using Tesseract and python

Figure 3.4 depicts the working methodology, initial the camera will be placed on top of the cluster in an isolated environment where no external disturbance will occur. Using provetech the test case is driven and respective image is captured by the camera, with fixed region of interest the text part alone will be cropped and preprocessed and further sent to Tesseract to extract string from image and gets the corresponding expected text from an excel sheet given by the customer, based on the matching the result will be stored in the excel sheet (pass/fail). In touch of a button the text verification for all the languages can be automated.

Also using Opencv preprocessing can be done for gray out images to test the string in it, with the correction in the first part of the process.


Fig 3.5 grade out text extraction using thresholding operation

By using thresholding operation the graded out text can be converted into lighter and using Gaussian blur the pixelated portions can be smoothed up and finally fed to Tesseract and will follow the same process as above as in fig 3.4.

### 3.2 Multi-Lingual Scrolling Text verification using Machine Learning algorithms:

This module utilizes the concept of panorama images that is nothing but Image key point stitching.

Fig 3.6 Panoramic Image stitching

For this technique Features are extracted using SIFT (Scale invariant Feature Transform) algorithm, where in which unique key point in a particular images will be extracted as feature and the keypoints are invariant to scale as shown in the fig 3.7 for vertical scrolling text and fig 3.8 for horizontal scrolling text.

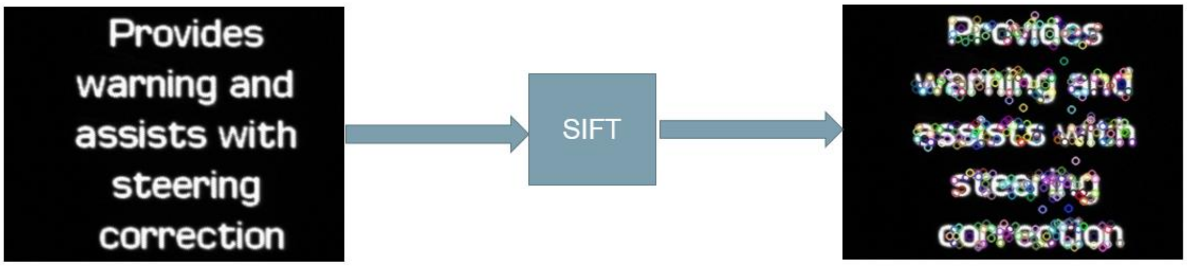


Fig. 3.7 vertical scrolling image feature extraction using SIFT


Fig. 3.8 Horizontal scrolling image feature extraction using SIFT

Once feature extraction is done, brute force matcher is used for identifying homogenous key point between two images as shown in the fig. 3.9 & 3.10

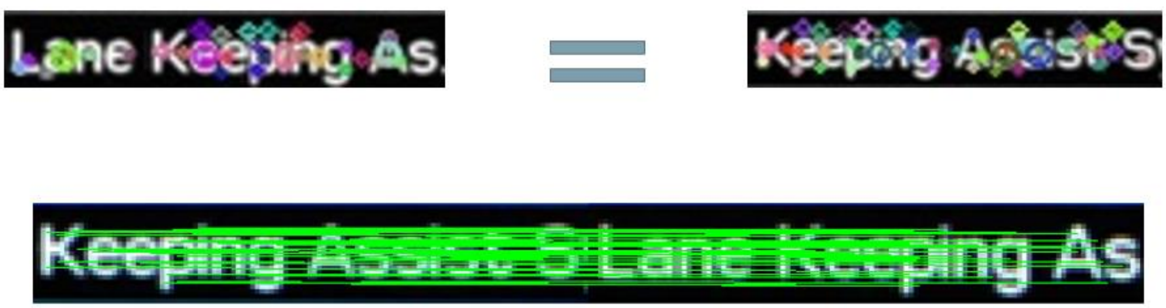


Fig. 3.9 Homogenous point matching using Brute force matcher for Horizontal scrolling image

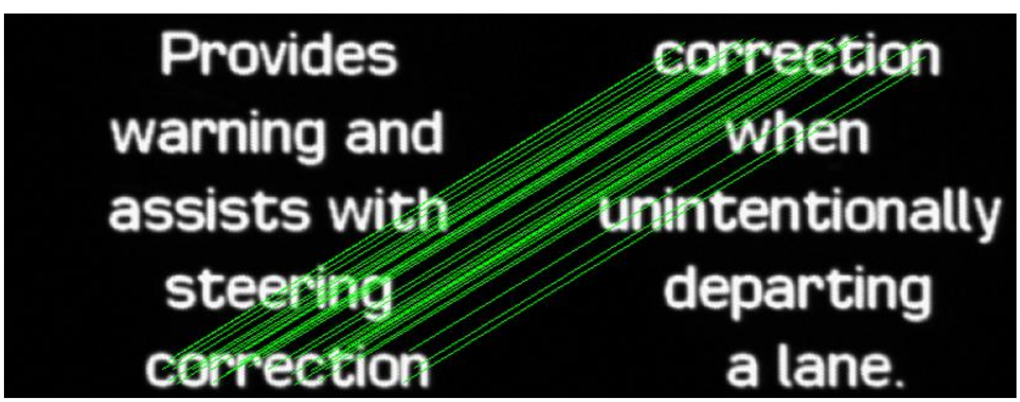


Fig. 3.10 Homogenous point matching using Brute force matcher for Vertical scrolling image

And finally when there are enough key point matched in two images it will be warped together to form a single image and likewise for all the successive images the same technique will be applied in such a way that the entire text can be addressed and a final image that has all the content in it. And the image will be fed to Tesseract OCR to extract string which is further validated with the expected string. The overall working of the concept is shown in the fig. 3.11.

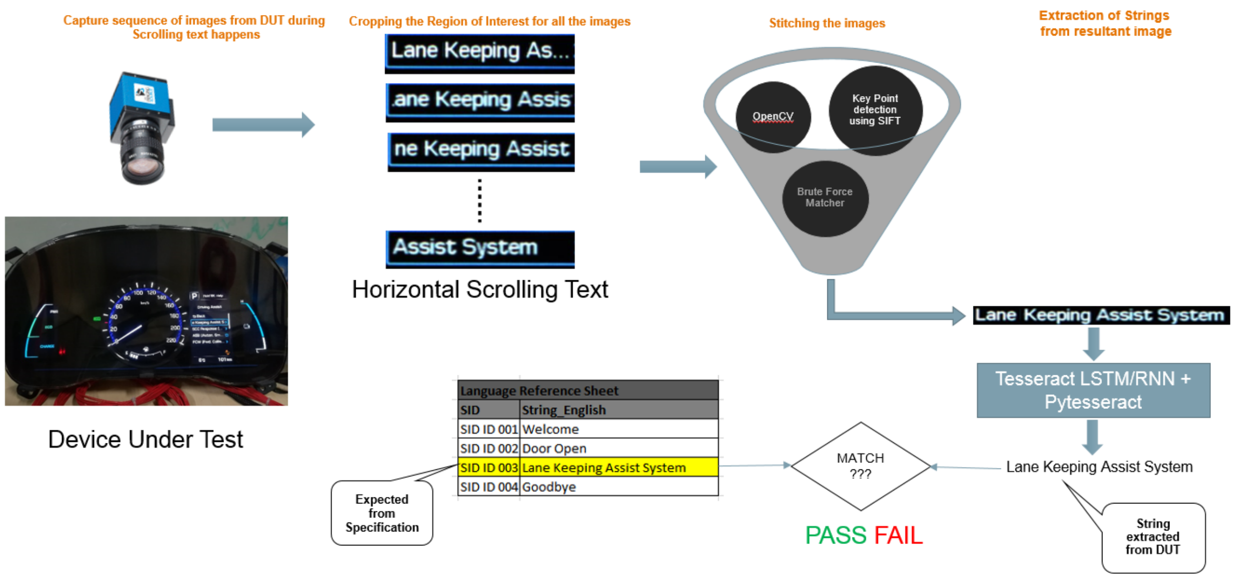


Fig. 3.11 Image Key point stitching for scrolling text verification using machine learning

Using this method all the scrolling text can be automated, as the size and speed of the scrolling is not a concern, as cameras takes 24 images per seconds which makes it more than enough to stitch it.

### 3.3 Auto text segmentation using Convolutional Neural Network

Segmentation of text from real time images is a very challenging part to address in the field of Artificial Intelligence. Text Segmentation is a technique used to extract text part alone in the real time images as shown in the fig. 3.12. The text segmentation is still under active research and is being implemented in self-driving cars.

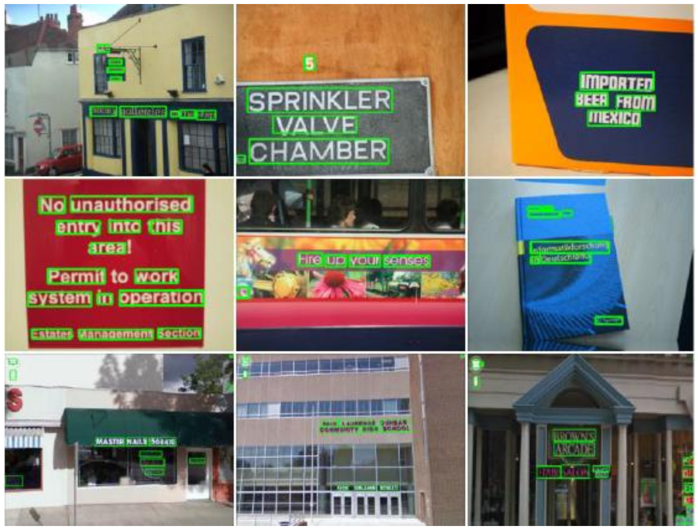


Fig. 3.12 text segmentation

In the OCR method, we are manually measuring the co-ordinates and given as a coordinates to the image such that only texts are cropped and sent to Tesseract for OCR, this will become more time consuming if there are more number of languages as the size of text will vary for each language, in order to mitigate this effect a deep learning model is created to identify the text regions in the image such that the region of interest can be found and cropped automatically and further validated using Tesseract OCR as shown in the fig. 3.13.

Fig. 3.13 Auto text segmentation using Convolutional Neural Network

In this technique we are applying series of image processing technique to minimalize the character candidates present in the image such that the model will be used very less for classifying the candidates belong to text or icon, the classification part is done using Inception V3 deep neural Network Architecture [8] as shown in the fig. 3.14.

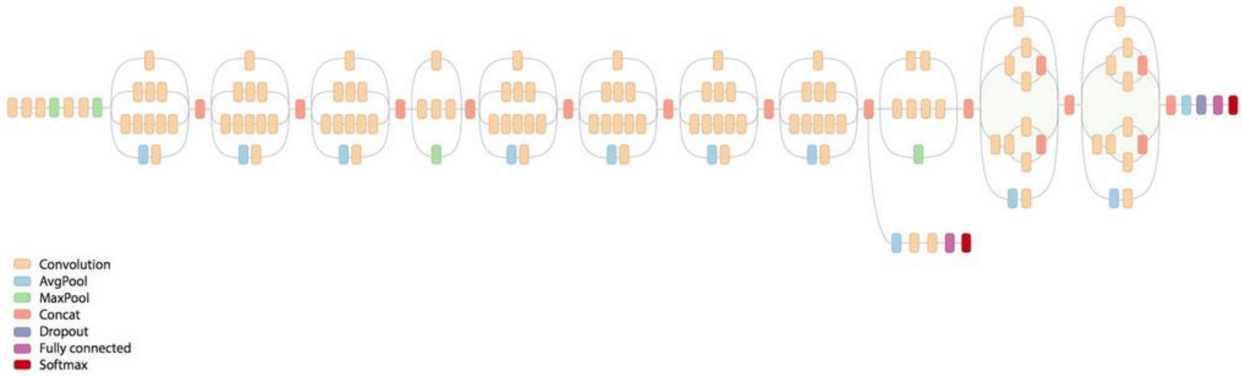


Fig. 3.14 Inception_V3 Deep Neural Architecture

So dataset is collected that particularly defines text and icons, later the features are extracted using inception v3 using keras as frontend Tensorflow as backend and logistic regression is used as the classification technique using scikit learn from which the final classifier model is generated in the form of cPickel format, using the model the given image can be classified as text/non-text as shown in the fig. 3.15.

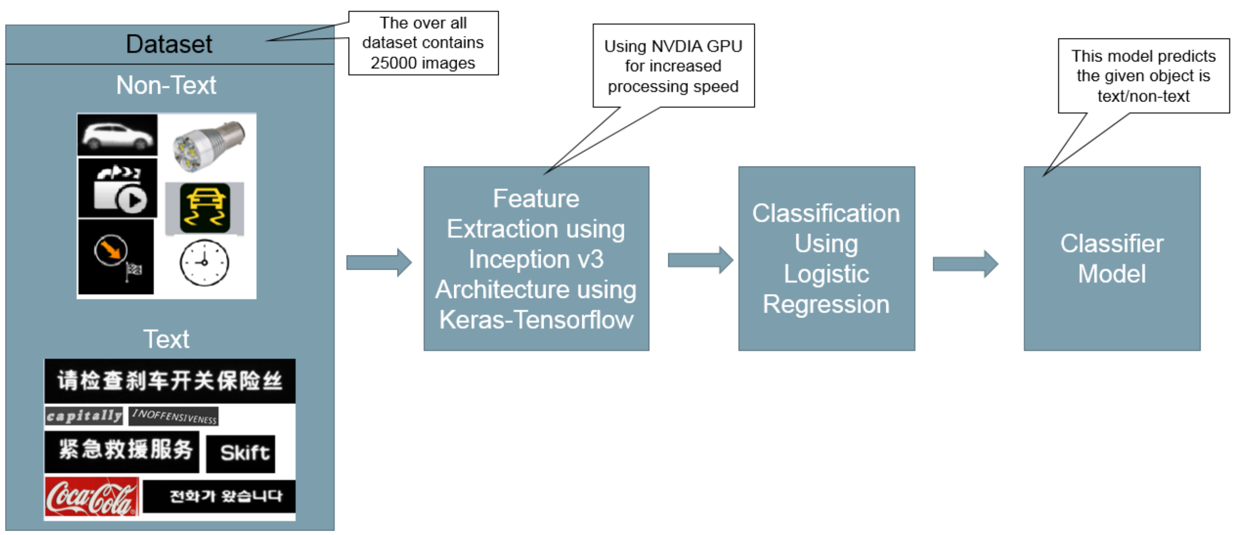


Fig. 3.15 Classifier model generation using Inception_v3 and Logistic Regression

And now the model is ready to classify the text/non-text regions, candidates of the images are cropped and send to the model, the classifier model will classify given images are text. For that series of steps are done to achieve it as shown in the fig. 3.16.

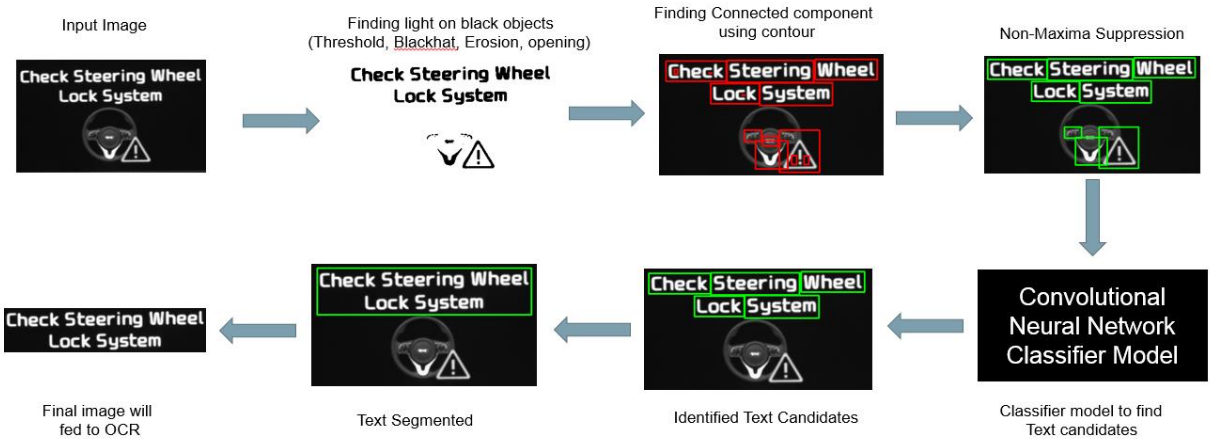


Fig 3.16 Text segmentation entire process

## 4. RESULTS & BUSSINESS BENEFITS:

Thus using the above mentioned techniques, end to end automation is achieved for string verification. We have utilized this testing tool in Hyundai Ionic Instrument Cluster which has 23 languages with 20,000 strings in it. Using these techniques the validation was done at ease for all the sprints which reduced the effort hours utilized and increase the accuracy that obliviously decreased the slip through rate to the customer. The below table Compares the effort hours reduction with the current and previous tools.

| Validation Task | Previous Method effort Hours | Current Method Effort Hours |
|---|---|---|
| Text verification in all the 23 languages | 90 | 30 |
| Scrolling text verification | 24 | 5 |
| Text Segmentation | 5 | 1 |

The technique can be applied for USM functionality (User Setting Menu) as there will be no feedback of where the focus bar is after some combination of key presses.

Also it can applied for Clock time verification, ODO and other number verification to.

## 5. REFERNCE

[1] http://www.ni.com
[2] https://www.abbyy.com
[3] https://github.com/tesseract-ocr
[4] https://opencv.org/
[5] http://tensorflow.org/
[6] http://scikit-learn.org/
[7] https://www.provetech.de
[8] http://www.how-ocr-works.com/
[9] Ekraam Sabir et. al. “Implicit Language Model in LSTM for OCR” at Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference, 2018.
[10] Christian Szegedy et. al. “Rethinking the Inception Architecture for Computer Vision” at Computer Vision and Pattern Recognition, 2015.