Detecting Handwritten Text from Forms using Deep Learning

Shailendra Singh Kathait
Co-Founder and Head of Analytics Valiance Analytics Pvt. Ltd.
Noida, Uttar Pradesh

Chirag Sehra
Data Scientist, Valiance Analytics Pvt. Ltd.
Noida, Uttar Pradesh

ABSTRACT
Digital Image Processing is an expeditiously emerging field possessing a large number of applications in science and engineering aspects. One of the most used applications in almost every sector is Optical Character Recognition (OCR). OCR is the electronic conversion of handwritten text into digital format which makes information processing from printed papers to data records easy, thus helping to electronically edit, search and store printed texts into machines. This text can then be used in variety of applications like machine translation, speech-to-text, pattern recognition etc.

OCR as a piece of software applies pre-processing to improve the recognition in images. This pre-processing step includes skewness correction, despeckling, layout analysis and line and word detection.

OCR saves tons of manual effort by recognizing handwritten text with word level detection resulting in an accuracy of 81% to 90%.

With form processing, one can capture information in digital format that can save time, labor and money. This helps in achieving a better accuracy in detection. Such systems range from minor application forms to large scale survey forms. Deep Learning algorithms dealing with computer vision related tasks can be used to build a recognition engine.

Keywords
Image Processing, Intelligent Character Recognition, Optical Character Recognition, Optical Mark Recognition, Form Handling.

1. INTRODUCTION
Image processing is being widely used as the data is growing enormously including image data. Images occupy more space than a variable or text. Image processing techniques can be employed to save details from such images and index them to some database. Character Recognition is automated extraction of data from handwritten forms available in various file formats. With the emergence of technology, deep learning has changed the entire landscape over the past few years. It has changed radically in the ways that it is interactable with humankind. Historically, computers performed deterministic task and algorithms which worked well in situations which involved elaborate calculations but failed at recognizing faces or answering questions to user queries. Deep Learning relies on neural nodes that are connected to each other in a network-like structure. It relies on multiple layers existing inside the neural network. With increase in resources, the number of layers started increasing in the deep learning networks. Deep learning algorithms started to emerge which could detect edges, and features at every layer of processing. With this evolution of layers, character level of recognition in handwritten text became possible. Character level recognition is not possible without prior pre-processing of the image by which feature could be detected in an image. Scale-Invariant Feature Transform (SIFT) is a feature detection algorithm that describes the detection of local features in an image. This algorithm was patented by David G. Lowe in 1999. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine transformation, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images [11]. Each layer of the deep learning network extracted different set of features. For example, in face recognition, first layer recognizes the edges in the image, the second layer recognizes facial features like ear or nose and the final layer recognized full faces. Thus, feature extraction from each layer became possible. Character level detection and layer-based feature extraction became base for the development of the proposed tool. While building this proposed tool, it was planned to use Google Vision's API for every text field in the image. But, transfer learning approach was used because Vision was not able to correctly determine the continuous region where numbers existed. This became the reason to prefer transfer learning models for digit extraction from the images. Customized modeling gives an edge of using own generated data for training and testing

2. LITERATURE OVERVIEW
There are various kinds of approaches that are used for text recognition and form processing. Different recognition methods are applied based on the visualization of pixels in the image. Distinct ways like neural network, pattern recognition etc. can be applied. Different kind of machine learning models offers distinct levels of accuracies. Google Cloud Platform offers computer vision based APIs by which pre-trained machine learning models are accessible via REST and RPC APIs. It provides a set of features for analyzing images. It can be used for character recognition from PDF/ TIFF and images with handwriting or text. The models used by Vision API services are always being improved in order to provide better recognition accuracy. Thus, accuracy of the model is variable and depends upon the type of image and the parameters sent to API. Deep Learning has emerged and worked very well for the ImageNet classification challenge [9] where it improved

from 26.2% error rate with SIFT to 15.3% error rate with AlexNet[5] and to 2.25% with Se-ResNet[12]. It has been a real game-changer in Artificial Intelligence. It has reached dominating stage in state-of-the-art object detection by crushing conventional image classification models. Google Vision provides APIs that are easy to integrate and require

less effort and reduces development time for developing similar kind of models. Google Vision thus reduces development and deployment effort and cost. Thus, it is a go-to choice for researchers and practical coders to get started with machine learning using Vision API. Vision API recognizes characters from dense documents easily, but the same is not true for sparse documents. Therefore, it is advised to pre-process the image and then supply to Vision API. Deep Learning methodology requires manual coding of neural networks, thus providing an option to change hyper-parameters, input and output types which makes it more suitable for tasks that require tweaking of neural network for distinct performance results. Thus, both the options become suitable choices for algorithms for our problem statement.

3. PROPOSED METHOD

All these forms are filled physically by the operator and are sent across via email, eventually decrepitating and blemishing the image. This introduces a lot of noise in form image. Due to this, algorithm has to be dynamic and hefty to cacophony. The form registration also has to be very accurate since the accuracy of the field image extraction and hence field recognition depends on it. [1]

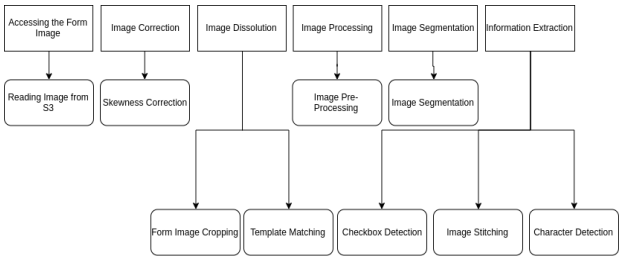

Figure 1: Proposed Methodology

3.1 Accessing the Image

The process starts with reading the image from the S3 location. A scanned image in jpg format is uploaded to S3 location. The image is scanned at 600dpi to ensure pixels of the characters in the image are clear to achieve maximum accuracy.

3.2 Image Correction

Image Correction involves correcting the asymmetry in the image. This is achieved by converting the image to grayscale format and then thresholding it. The image is thresholded with binary filter, a rotated rectangle of the minimum area in the image is found followed by calculation of an affine matrix of 2D rotation. The image is then rotated in the opposite direction from the base angle which straightens the image.

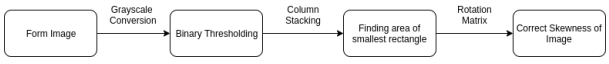

Figure 2: Image Correction

3.3 Image Dissolution

3.3.1 Image Cropping

The form image is converted to binary format. All the vertical and horizontal lines are extracted from the image and a mesh like structure is created. Contour detection is used to extract the contours from the processed image. The contour with the largest area is extracted out which is the bounding box around main entries.

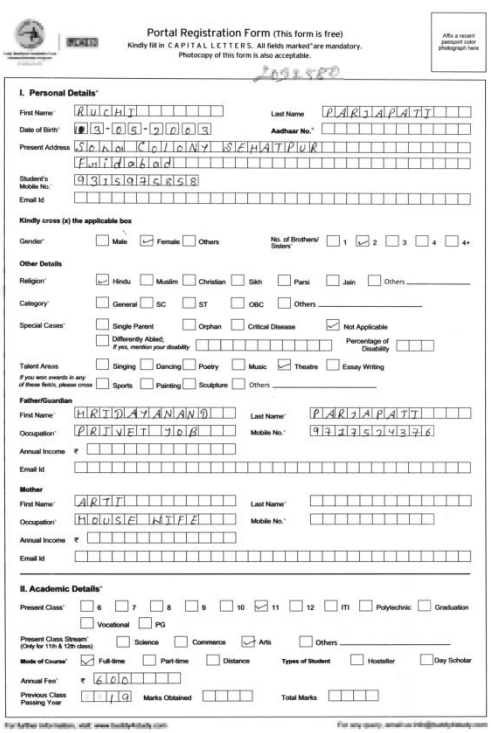

Figure 3: Complete Form Image

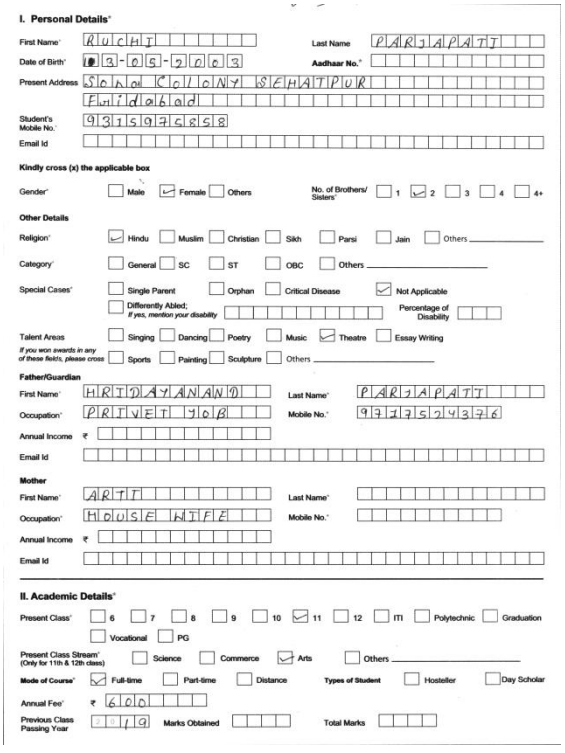

Figure 4: Cropped Form Image

This cropped portion is saved and is used as the new base form image.

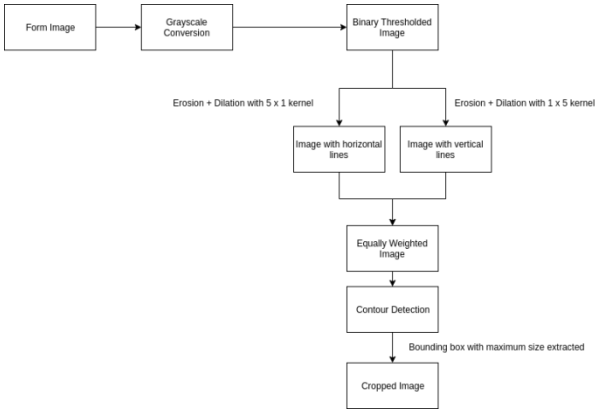

Figure 5: Image Cropping

3.3.2 Template Matching

XML is a text-based markup language which uses tags to store and organize data. An XML file is created both for the front and back side of the form. XML file contains the information about the boxes in the image and the coordinates associated with them. It also contains the meta-information of the image like the width and height of the image, its path and name. This well-formulated structure property of XML allow to easily extracting information. Thus, a tree-like-structure is created with all the basic information.

```
<annotation>
    <folder>forms</folder>
    <filename>form_01.jpg</filename>
    <path>/studybuddy/template/form_01.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>4500</width>
        <height>6000</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
<object>
    <name>first_name_box_0</name>
    <pose>Unspecified</pose>
    <folder>first_name</folder> <truncated></truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>667.0</xmin>
        <ymin>265.0</ymin>
        <xmax>776.0</xmax>
        <ymax>355.0</ymax>
    </bndbox>
</object>
<object>
    <name>first_name_box_1</name>
    <pose>Unspecified</pose>
    <folder>first_name</folder>
    <truncated></truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>784.0</xmin>
        <ymin>265.0</ymin>
        <xmax>893.0</xmax>
        <ymax>355.0</ymax>
    </bndbox>
</object>
```

Figure 6: XML Structure

3.4 Image Processing

Image processing involves various steps by which images are converted to a format that is easily processed by the algorithm. It involves noise removal, erosion and dilation of image. The main aim of pre-processing is to improve the image quality so that features of the image are enhanced. Morphological Operations and geometrical transformations are also performed while processing the image.

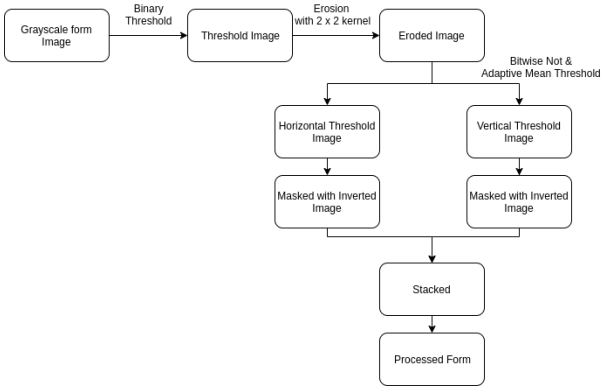

Figure 7: Image Processing

The cropped form image is thresholded into binary format. It is then eroded with a 2x2 kernel. Adaptive Mean Threshold is used for thresholding the image. A horizontal kernel is created with specified erosion and dilation of the image. Similarly, a vertical kernel is also created. These kernels are then used for producing masked images. These masked images are then stacked onto each other. It is then followed by a process of advanced morphological transformations.

3.5 Image Segmentation

Image Segmentation refers to the procedure of segmenting the images from the pre-processed image. There are various categories of segmentation which can be used [1]:

1) **Threshold based segmentation**: Histogram thresholding and slicing techniques are used to segment the image. They may be applied directly to an image, but can also be combined with pre-processing and post-processing techniques.

2) **Edge based segmentation**: With this technique, detected edges in an image are assumed to represent object boundaries and used to identify these objects.

3) **Regional based segmentation**: Where an edge based technique may attempt to find the object boundaries technique and then locate the object itself by filling them in, a region based technique based on the opposite approach, by starting in the middle of an object and then "growing" outward until it meets the object boundaries.[2]

There are two types of fields in our input.

1. Text based fields
2. Checkbox fields

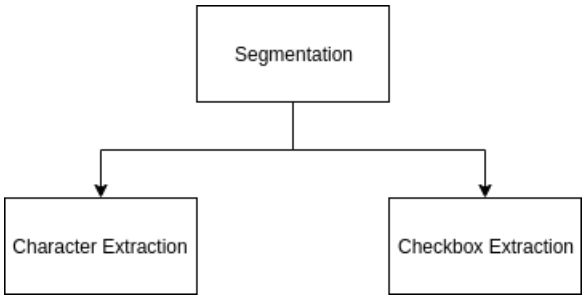

Figure 8: Image Segmentation Methodology

With the structure created in template matching step, each field in the image (irrelevant of its type) is segmented into folder structure with folder name as the field name and having all the entries of each box of the entries as the files of that folder. For example: *first name* field is extracted from the form and each letter of the first name is segmented in a folder named first name in the directory.

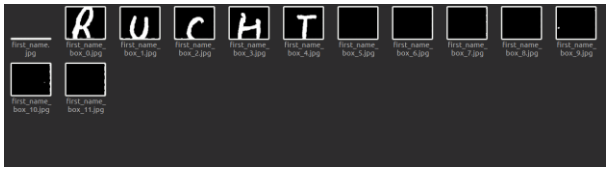

Figure 9: Content inside a segmented folder

3.6 Information Extractor

Information extractor piece of this software helps in extracting out the handwritten characters from the image to a digital format.

3.6.1 Checkbox Detection

Checkbox Detection refers to extracting the information from the checkbox field present in the form image. Since, all the fields are segmented down and stored into a directory, checkbox fields becomes easily accessible. All the available options of the checkbox fields are fed into the algorithm and the pixel values of each image is analyzed. Pixel count in the image can give us an idea of the stroke made in the checkbox. As an example, an '*x*' in the checkbox will have more amounts of white pixels rather than an image with no '*x*' marked. Thus, total number of white pixels in the image is calculated and fields are marked where the checkbox value has white pixels over a specific threshold value. This way, information from the checkbox field is extracted.

3.7 Image Stitching

Image stitching refers to the process of combining images in order that form up a complete field value. Each box inside any field has specific dimensions of 90 x 109 pixels. All the folders which are text fields are parsed and all the images inside the folders are stitched in orderly fashion and are saved in the same folder

3.8 Character Detection

There are various methods available for detecting text and handwritten characters with machine learning. We can use custom built model from scratch, use transfer learning for detection or use any available API over the web like Google Vision, AWS Rekognition etc. After extensive research and comparing the available options, the approach to detect text fields with Google Vision API was found. This approach is well suited for images and documents with dense text and using transfer learning for detecting fields which contains numeric digits. Google's Inception v3 model is used for detecting numbers from numeric fields like mobile number and Aadhar Identification Number. Google's Inception model is basically a deep neutral network which was used for classification in ImageNet's classification challenge back in 2012.

ANNs work similar to a human brain. Human brain constitutes of 100 billion neurons. All these neurons are then connected to other axon cells. Dendrites accept the inputs from sensory organs which created electric signals which quickly travel through the neural network. Neurons are connected to each other by links for interaction. These neural nodes can perform operations on data they intake. Outgoing like of the node transfers the output of operations performed to the next node. The output at each node has an activation associated with it. There are weights associated with each link between the nodes. Thus, a neural network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs. [3]

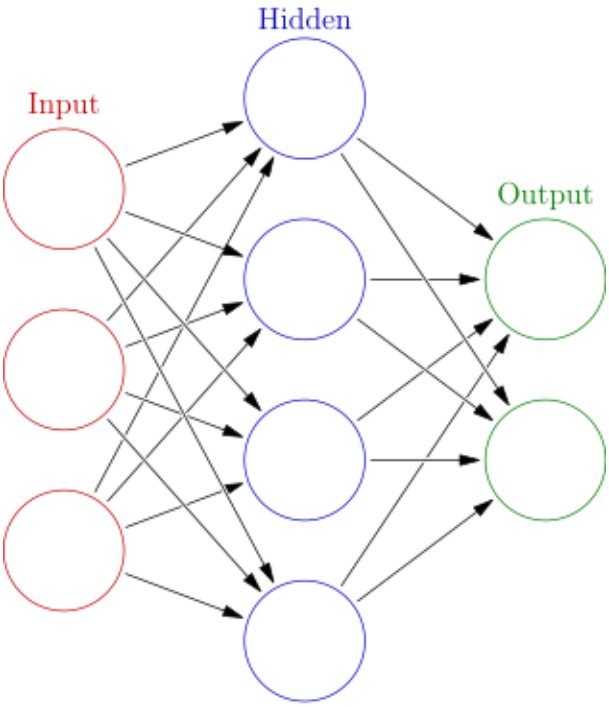

Figure 10: Basic Neural Network, Source: Adapted from [16]

Artificial Neural Network (ANN) ANN is essentially an information processing hub that works in similar fashion like the biological nervous system. There are large numbers of interconnected processing nodes that works together to solve a problem.

Some of the features of Artificial Neural Networks are:

1. **Modular Learning**: Ability to learn on how to do tasks based upon the data provided for training experience.
2. **Real Time Operation**: ANNs can carry out calculations in parallel, in multi-core architecture.
3. **Fault Tolerance**: Major ANNs can retain the operations when major damage in the network may happen.

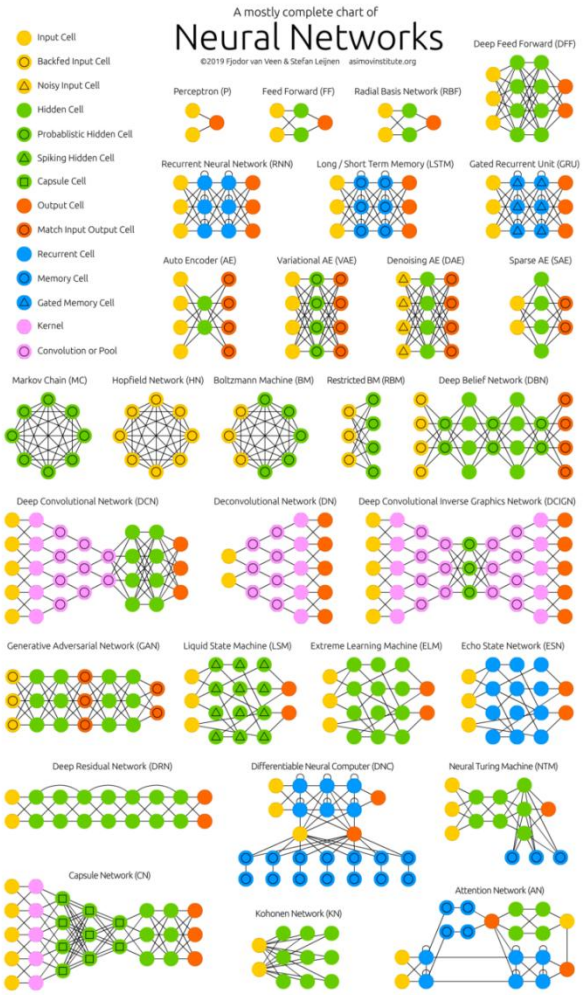

Figure 11: The Neural Network Zoo, Source: Adapted from [17]

There are various kinds of neutral networks that can be built by changing the structure of nodes.

1. **Feed-Forward Neural Network(FNN)**
   These networks are straight forward. They feed information to each node from the input layer to the output layer. Each layer consists of input, hidden and output layer in parallel. Each neuron from one layer is connected to every neuron in the next layer. FFNNs are generally trained through back-propagation. The error i.e. variation of difference between the input and output is back-propagated to the network.

2. **Recurrent Neural Networks (RNN)**
   RNNs are FNNs that take state into account. Neurons in the network are fed with information from previous layer as well as from the previous pass. RNNs exhibit exploding gradient problem where the activation function rapidly losses information over time. These are well suited for problems that have a sequence associated with them.

3. **Deep Learning Algorithms**
   Deep learning is the branch of machine learning which is based on the fact that model has a deep graph with multiple layers that compose of multiple linear and non-linear transformations. Deep Learning algorithms can automatically learn feature representation making them the state-of-the-art solution to beat problems. These algorithms are based on large neutral networks that are inspired by the human brain.

For detection of handwritten text, the best neural network architecture is of Convolutional Neural Networks. There are various libraries that are available for completing the task like Theano, NeuPy, PyTorch and Tensorflow. Out of all the available options, Tensorflow is the best suited as the neural network can be loaded into memory and images can be processed in distributed environment.

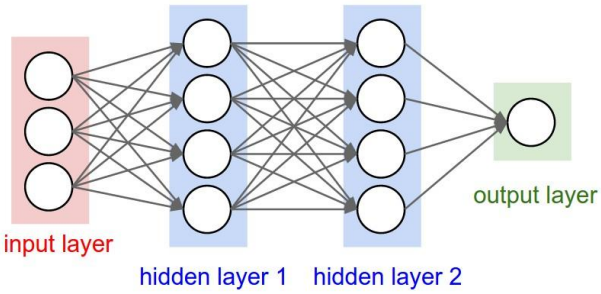

Figure 12: Layered Convolutional Neural Network, Source: Adapted from [18]

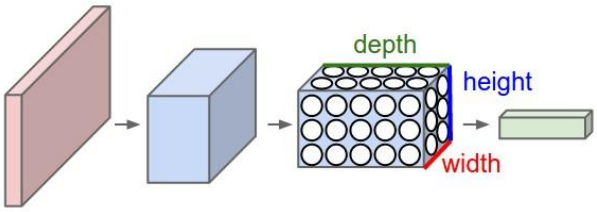

Figure 13: Convolutional Neural Network, Source: Adapted from [18]

1. Convolutional Neural Networks:
   Convolutional Neural Networks (ConvNet) are structurally similar to vanilla neural networks. Each node intakes, performs mathematical operation like dot product and then performs a non-linear function on it. This network translates an image with raw pixels to the class predictions on the other end. It is also associated with a loss function on the last layer of the network (e.g. softmax, mean square error etc.). Neural Networks intakes a vector, applies transformation with a series of hidden layers. Each hidden layer constitutes of neuron nodes that are fully connected to all nodes in the predecessors' layer. The fully connected layer at the end is called '*output layer'*. ConvNet are used for processing on images because unlike a neural network, the layers of network are arranged in 3 dimensions: width, height and depth. Each node in a CNN receives an input and performs a set of linear and non-linear operations. ConvNet are majorly used for pattern recognition in images.

A ConvNet consists of 3 types of layers:

1. Convolutional Layer: Convolutional layer is the primary layer that extracts features for an input image. It preserves the association

between pixels and image features. This is done by using small squares of input information. During a forward run, each small square or filter is convolved over the complete image and computing the dot product between entries of filter and input thus creating a 2D map of the filter.

2. Pooling Layer: Pooling layer is added after the convolutional layer has been applied with a non-linear activation function. It involves selecting a pooling operation, like a filter. Pooling layer reduces the size of each feature map. The pooling operation is specified, rather than learned. Two common functions used in pooling operations are Average Pooling and Maximum Pooling (or Max Pooling).

   Average Pooling is used to calculate the average value of each patch on the feature map. Max Pooling is used to calculate the maximum value for each patch of the feature map. Pooling creates a summarized version of featured in the input. They are useful as small changes in the location of the feature detected by the convolutional layer results in a pooled feature map with the feature in the same location. In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change. [4]

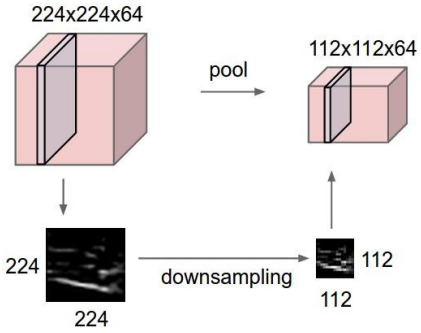

Figure 14: Pooling Layer Functionality, Source: Adapted from [18]

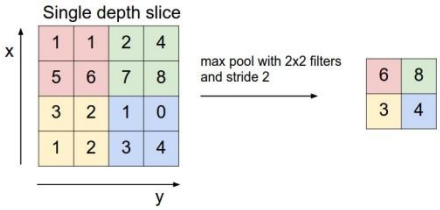

Figure 15: Pooling Layer Functionality, Source: Adapted from [18]

3. Fully Connected Layer: After various layers of convolutional and max pooling, the main reasoning starts with fully connected layers. Nodes in a fully connected layer have connections to activation in the previous layer. These activation functions are computed with matrix multiplication operations and add a bias offset with it. [5]

With the current problem statement, we have a classification problem, where we need to predict the output class on basis of previously trained data-set.

1. Tensorflow: Tensorflow is an end-to-end open source platform for machine learning which has a comprehensive, flexible ecosystem of tools and libraries that let researchers push the state-of-the art in ML build and deploy ML powered applications. ML models can be built and trained easily using the high-level APIs with keras which makes the model iteration and debugging easy.

   Tensorflow does numerical computation using data flow graphs. Nodes in the graph represent mathematical operations and the edge denotes the multidimensional tensors (data arrays). Tensorflow's architecture allows deploying computational graphs on one or more CPUs and GPUs. [10]

3.8.1 Text Field Detection

All the text fields taken into account are alphabetical and alphanumeric characters. The Vision API uses *DOCUMENT_TEXT_DETECTION* to detect handwriting from an image or file. The JSON response contains several information like the page, paragraphs, blocks, words and information about where a break occurs in the dense text in image. There are several languages that Google Vision supports. By default, Vision API uses automatic language detection.

Google vision requires calling an *ImageAnnotatorClient* method with service account credentials passed to it as parameters of the method. All the fields are recursively read from the folder structure created and the JSON response is intelligently parsed and the information required is stored in another JSON object which contains the *text_field_name*, *word_text* and *word_confidence*.

3.8.2 Numeric Field Detection

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [13]. Pre-trained models are used as an origin point as these are trained with immense resources that are primarily used to train deep learning models. For the given task, we used Google's Inception v3 model [6]. Inception v3 exhibits greater than 78.1% accuracy on the ImageNet dataset [9]. The model constitutes of symmetric and asymmetric sections and has various layers like convolutions, max pooling, average pooling, dropouts, concatenations and fully connected layers. Batch Normalization is extensively used to activate the inputs to the layers. Inception v3 used softmax as the loss function for its core network layers.

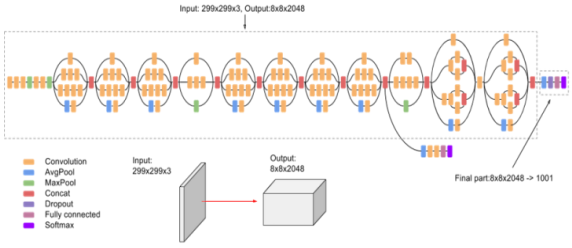

Figure 16: Inception v3 Architecture, Source: Adapted from [19]

We fine-tuned the pre-trained model for a detection of numeric digits with specific dimensions. This process requires building the exact same model where we change the number of labels in the final classification layer and restore all weights from the pre-trained Inception-v3 model except the final classification layer. Inception-v3 has better factorization resulting in smaller convolutions. For example, single layer of 5x5 filters will have 25 parameters, while 2 layers of 3x3 filters will have 18 parameters. Here, the number of parameters is reduced by 28%. The factorization is done on symmetric convolutions. This factorization could also be done producing asymmetric convolutions, i.e. 3x3 convolutions being replaced by 3x1 convolution followed by 1x3 convolutions. A single 3x3 convolution produces 9 parameters while 3x1 and 1x3 filters, produces 6 parameters thus number of parameters reduces by 33%. There are a total of 42 layers in the model, and the computation cost is 2.5 times higher than that of GoogLeNet [7] and much more efficient than VGGNet [8]. The primary dataset for training has 40000 images and testing dataset has 10000 images in total of all images. All the images are segregated based on the digits in the image. 'Bottleneck' refers to the layer just before the final output, before the actual classification layer also called as image feature vector. It is trained to an output set of values that is good enough for classifier to distinguish images for the problem here. Re-training the final layer can work on our new classes as the network was originally used to distinguish between 1000 classes in ImageNet.

Tensorflow Key points:

1. The standard image size used for training and testing purposes is 90 x 109 pixels equivalent to 9810 different features.
2. Variation in range of pixel values of the image is between 0 and 255 pixels.
3. The number of iterations are increased and adjusted as per the increase in accuracy of the model.
4. One hot vector of the target labels are fed into the network.
5. Complete dataset is divided into training and testing data in ratio of 80: 20.
6. The training data consists of handwritten samples of digit fields only i.e. from 0 to 9.
7. Weights of different models received after training network are saved.
8. Bottleneck weights of the network are available and a final classification layer with 10 classification neuron nodes are added as a layer. This layer is trained over the bottleneck structure.
9. The weights and biases are adjusted according to the dataset.
10. Training is done on 100 epochs for 5 batches.
11. The model is saved and tested on test-data set.

3.9 Process Flow of Application

The system is used as an application program interface (API) by any client-side application. API is a set of routines and protocols that can interact with graphical GUI components to serve the client application with a backend system. This API is developed with Django. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.[14]. This API is deployed on Amazon Elastic Compute Cloud instance (Amazon EC2). Amazon EC2 is a web service that provides secure, re-sizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment [15]. Data can be retrieved or saved with help of any API by sending in the correct form of request. Client application generates a request to the API sending in the s3 link of the form image along with an 'id'

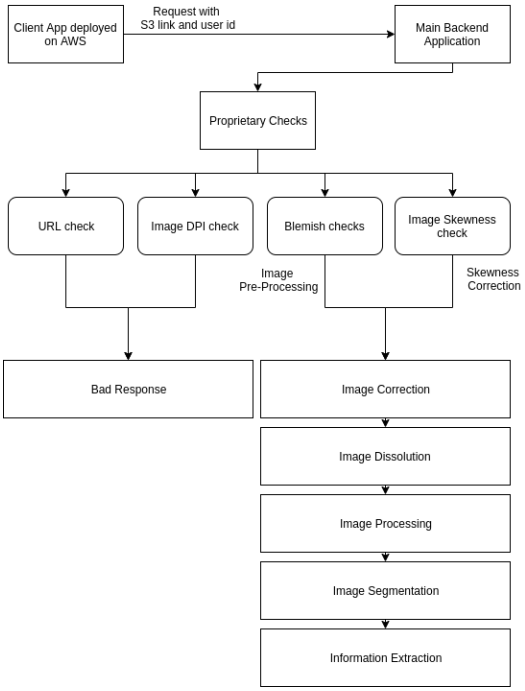

Figure 17: Process flow of Application

which is unique and the 'form side' for that request. The discussed module runs over the EC2 instance and generates response containing information whether the complete process was successful or whether it failed. In case of any failure, associated error codes are returned along within the response. They are 4 possible situations in which a failure could happen:

1. Rectifiable Failures: These failures are handled and can be rectified by employing specific techniques.
    - Image Skewness: While scanning, the forms may get a bit rotated which could cause problem while detection. This problem is handled in the Image Skewness Correction section where angle of skewness is calculated and form is rotated in the opposite sense of direction.
    - Unwanted Blemished and lines: While forms are collected, there might be situation where forms might get crumbled or there might be sections of blemishes in the form image. This problem is handled in the Image Pre-processing section where erosion and dilation is done along with color conversions in the image.

2. Non- Rectifiable Failures:

    - Incorrect URL: Client application might sometimes send a bad URL to the API on EC2 instance. Such a problem is not easy to handle and thus an error code of 'BAD URL' is returned in the in the response.
    - Form Image DPI check: Forms are manually scanned and there is a possibility that while scanning, someone could scan the form at a lower resolution than 600 dpi. This could lead into a situation where forms are not scaled correctly as per the XML template. Therefore, to avoid such a situation a DPI checks if applied over the image that is passed in the request. A '*BAD DPI SCAN*' error code is returned in the response.

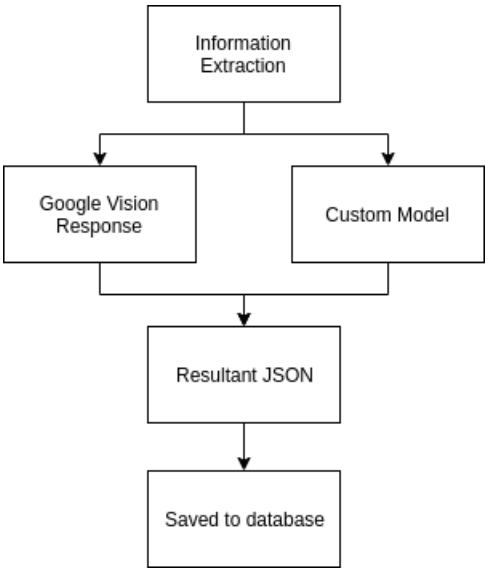

Figure 18: Information Extraction and Saving to Database

Once the image is corrected, it undergoes the same procedure of Image Dissolution, Image Processing, Image Segmentation and Information Extraction. A JSON response is returned from the Google Vision's API containing the text present in the image along with its accuracy. Custom model built with transfer learning also returns JSON object with digits in the image along with the accuracy. Both the results are combined and is saved into a SQL database. Information of any of the user can then be queried using the unique id that is present in the request object.

4. CONCLUSION

This discussed approach works well for the handwritten forms available. The accuracy of these models can be increased further from 90 by increasing the amount of training dataset. The recognition tool built can be made to use for any type of handwritten/printed form of any kind and the data can be saved in the database in much less time as compared to manually entering the data.

5. REFERENCES

[1] Shailendra Singh Kathait and Shubhrita Tiwari. Application of Image Processing and Convolution Networks in Intelligent Character Recognition for Digitized Forms Processing. Valiance Solutions Pvt. Ltd., Noida, Uttar Pradesh, 2018

[2] Dipti Deodhare, NNR Ranga Suri and R. Amit. Preprocessing and Image Enhancement Algorithms for a Form-based Intelligent Character Recognition System. Technomathematics Research Foundation, 2005

[3] Maureen Caudill. Neural Network Primer: Part I. AI Expert, Feb. 1989

[4] Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning MIT Press, http://www.deeplearningbook.org, 2016

[5] Alex Krizhevsky, Geoffrey E. Hinton and Ilya Sutskever. ImageNet Classification with Deep Convolutional Neural Networks. University of Toronto.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna Rethinking the Inception Architecture for Computer Vision https://arxiv.org/abs/1512.00567

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich Going Deeper with Convolutions

[8] Karen Simonyan, Andrew Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition https://arxiv.org/abs/1409.1556

[9] ImageNet: 2016 Stanford Vision Lab, Stanford University, Princeton University http://www.image-net.org/

[10] Tensorflow https://www.tensorflow.org/

[11] David G. Lowe Distinctive Image Features from Scale-Invariant Keypoints https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[12] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu Squeeze-and Excitation Networks https://arxiv.org/abs/1709.01507

[13] Lisa Torrey and Jude Shavlik Transfer Learning, Handbook of Research on Machine Learning Applications, 2009. http://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf

[14] Django https://www.djangoproject.com/

[15] Amazon Web Services https://aws.amazon.com/ec2/

[16] Artifical Neural Network https://en.wikipedia.org/wiki/Artificial neural network.

[17] Fjoder Van Veen The Neural Network Zoo https://www.asimovinstitute.org/neural-network-zoo/

[18]]karpathy@cs.stanford.edu CS231n: Convolutional Neural Networks for Visual Recognition http://cs231n.github.io/convolutional-networks/

[19] Google Cloud https://cloud.google.com/tpu/docs/inception-v3-advanced