1. INTRODUCTION

Over recent years, the landscape of computer vision has been drastically altered and pushed forward through the adoption of a fast, scalable, end-to-end learning framework, the Convolutional Neural Network (CNN). Though not a recent invention, we now see a cornucopia of CNN-based models achieving state-of-the-art results in classification, localisation, semantic segmentation, and action recognition tasks, amongst others. A desirable property of a system which is able to reason about images is to disentangle object pose and part deformation from texture and shape. The introduction of local max-pooling layers in CNNs has helped to satisfy this property by allowing a network to be somewhat spatially invariant to the position of features. However, due to the typically small spatial support for max-pooling (e.g. 2 × 2 pixels) this spatial invariance is only realised over a deep hierarchy of max-pooling and convolutions, and the intermediate feature maps (convolutional layer activations) in a CNN are not actually invariant to large transformations of the input data. This limitation of CNNs is due to having only a limited, pre-defined pooling mechanism for dealing with variations in the spatial arrangement of data.

In this work we introduce a Spatial Transformer module, that can be included into a standard neural network architecture to provide spatial transformation capabilities. A spatial transformer network is a specialized type of convoluted neural network,or CNN. Spatial transformer networks contain spatial transformer modules that attempt to make the network spatially invariant to its input data. In essence, a spatial transformer network is used when attempting to stabilize, or clarify an object within a processed image or video. This leads to more accurate object classification and identification.

1.1 MOTIVATION FOR WORK

Our motivation is to use these capabilities of CNNs and create an end-to-end scene text recognition system that behaves more like a human by dividing the task at hand into smaller subtasks and solving these subtasks independently from each other. In order to achieve this behaviour, we learn a single DNN that is able to divide the input image into subtasks (single characters, words or even lines of text) and solve these subtasks independently of each other.

1.2 PROBLEM STATEMENT

The main aim of the proposed method is to detect the characters from the given natural images using Convolutional neural network (STN).

This project is aimed at developing software which will be helpful in recognizing characters of English language. This project is restricted to English characters only. It can be further developed to recognize the characters of different languages. It engulfs the concept of neural network. One of the primaries means by which computers are endowed with humanlike abilities is through the use of a neural network.

2.2 EXISTING METHODS

2.2.1 Text Localization and Recognition in Real-World Images

A general method for text localization and recognition in real-world images is presented. The proposed method is novel, as it (i) departs from a strict feed-forward pipeline and replaces it by a hypotheses-verification framework simultaneously processing multiple text line hypotheses, (ii) uses synthetic fonts to train the algorithm eliminating the need for time-consuming acquisition and labeling of real-world training data and (iii) exploits Maximally Stable Extremal Regions (MSERs) which provides robustness to geometric and illumination conditions.

The performance of the method is evaluated on two standard datasets. On the Char74k dataset, a recognition rate of 72% is achieved, 18% higher than the state-of-the-art. The paper is first to report both text detection and *recognition* results on the standard and rather challenging ICDAR 2003 dataset. The text localization works for number of alphabets and the method is easily adapted to recognition of other scripts.

2.2.2 Reading Digits in Natural Images with Unsupervised Feature Learning

Detecting and reading text from natural images is a hard computer vision task that is central to a variety of emerging applications. Related problems like document character recognition have been widely studied by computer vision and machine learning researchers and are virtually solved for practical

to existing scene text recognition methods: (i) It can recognise highly ambiguous words by leveraging meaningful context information, allowing it to work reliably without either pre- or post-processing; (ii) the deep CNN feature is robust to various image distortions; (iii) it retains the explicit order information in word image, which is essential to discriminate word strings; (iv) the model does not depend on pre-defined dictionary, and it can process unknown words and arbitrary strings. It achieves impressive results on several benchmarks, advancing the-state-of-the-art substantially

2.2.6 Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks

Recognizing arbitrary multi-digit numbers from Street View imagery. Traditional approaches to solve this problem typically separate out the localization, segmentation, and recognition steps. In this paper we propose a unified approach that integrates these three steps via the use of a deep convolutional neural network that operates directly on the image pixels. We employ the DistBelief (Dean et al., 2012) implementation of deep neural networks in order to train large, distributed neural networks on high quality images. We find that the performance of this approach increases with the depth of the convolutional network, with the best performance occurring in the deepest architecture.

3. METHODOLOGY

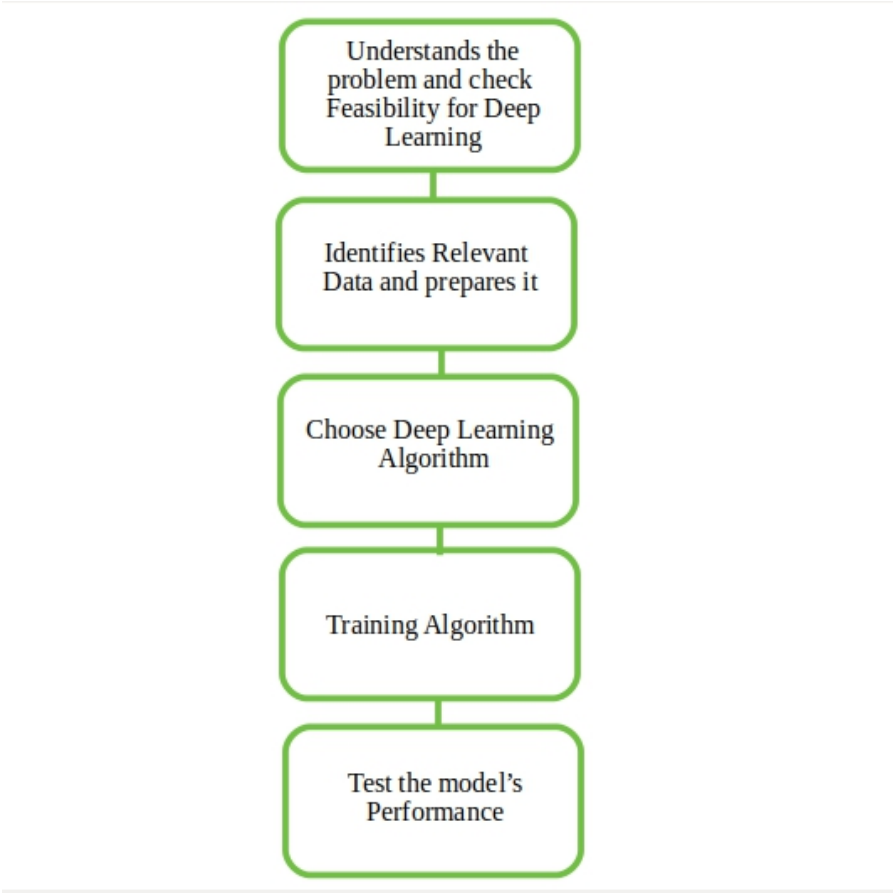
3.1 Proposed system

A human trying to find and read text will do so in a sequential manner. The first action is to put attention on a line of text, read each character sequentially and then attend to the next line of text. Most current end-to-end systems for scene text recognition do not behave in that way. These systems rather try to solve the problem by extracting all information from the image at once. Our system first tries to attend sequentially to different text regions in the image and then recognize the textual content of each text region.

In order to this we created a simple DNN consisting of two stages: (1) text detection (2) text recognition. In this section we will introduce the attention concept used by the text detectionstage and the overall structure of the proposed system. We also report best practices for successfully training such a system.

3.2 Working:

First, we need to identify the actual problem in order to get the right solution and it should be understood, the feasibility of the Deep Learning should also be checked (whether it should fit Deep Learning or not). Second, we need to identify the relevant data which should correspond to the actual problem and should be prepared accordingly. Third, Choose the Deep Learning Algorithm appropriately. Fourth, Algorithm should be used while training the dataset. Fifth, Final testing should be done on the dataset.



ALGORITHM

In traditional image processing field, rotational invariance or scale invariance is of great importance, and actually, there are many feature descriptors such SIFT and SURF famous for their consistent performance against affine transformation. In the era of Deep Learning, we tend to trust neural network could handle everything for us automatically, but I figure it is the point why a considerate amount of people don't value this method.

Objectively, the design of CNN could be insensitive to some slight rotation or translation transformation. For example, pooling layer, could tolerate the pixel switch inside the pooling window. But the following distorted mnist image may challenge the capability of CNN to extracting the most salient feature in the image.

3.3 SPATIAL TRANSFORMATION NETWORK

STN model behaves like a human, it will start reading line by line in sequential manner and read each character step by step. These systems perform operations on complete image and extract all information at once. In this study, human-based approach is followed to find and localize textual regions sequentially in images and then recognize those localized textual regions. In this regard, Neural Network model is developed which is comprised of two stages: 1) text detection and 2) text recognition.

.

our system we use the spatial transformer as the first step of our network. The localization network receives the input image as input feature map and produces a set of affine transformation matrices that are used by the grid generator to calculate the position of the pixels that shall be sampled by the bilinear sampling operation.

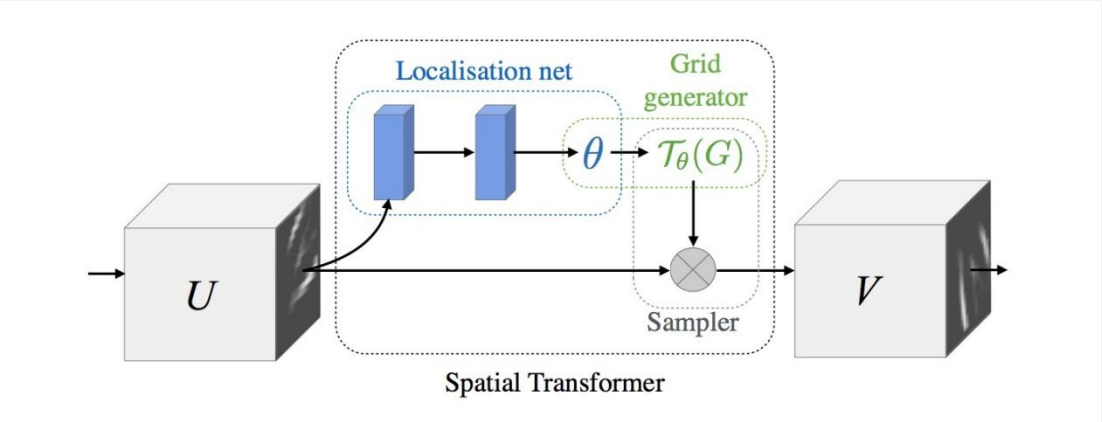
3.5 TEXT RECOGNITION STAGE:

The image sampler of the text detection stage produces a set of N regions that are extracted from the original input image. The text recognition stage uses each of these N different regions and processes them independently of each other. The processing of the N different regions is handled by a CNN. This CNN is also based on the ResNet architecture as we found that we could only achieve good results if we use a variant of the ResNet architecture for our recognition network. We argue that using a ResNet in the recognition stage is even more important than in the detection stage, because the detection stage needs to receive strong gradients from the recognition stage in order to successfully update the weights of the localization network. The CNN of the recognition stage predicts a probability distribution yˆ over the label space Lǫ, where Lǫ = L U {ǫ}, with L = {0 − 9a − z} and ǫ representing the blank label. Depending on the task this probability distribution is either generated by a fixed number of T softmax classifiers, where each softmax classifier is used to predict one character of the given word:

$$x^n = O^n$$
$$\hat{y}_t^n = softmax(f_{rec}(x^n))$$
$$\hat{y}^n = \sum_{t=1}^{T} \hat{y}_t^n$$

From input to output

This equation tells us, only using six parameters could we define an affine transformation and our goal comes out quite obviously **that for each image, our model could output six parameters from one of its layers and these parameters decide how the [x^t, y^t] should be transformed to [x^s, y^s]**(e.g rotation, shift, scale). The above work is done by the layer called **Localisation Net.**



More formally, the Localisation net is defined as follows:

- **input**: feature map U of shape (H, W, C)
- **output**: transformation matrix θ of shape (6,)
- **architecture**: fully-connected network or ConvNet as well.

Here we come down with another key point when dealing with this kind of transformation tasks, which matters but tends to be ignorant unless being implemented. For example, what we wanna get a transformed image of 100*100 from an original image of 100*100, mapping function is **F**. Suppose **(x,y)** with **g** grayscale in the original image, and **(x',y')** in the transformed image is