Capstone Proposal

Machine Learning Engineer Nanodegree

Pushkar Kurhekar

01/02/2021

Domain Background

For Optical Character Recognition (OCR) tasks, quality of the input document is directly proportional to the output accuracy of the text. Quality problems in the input document images such as skew, perspective transformation, watermarks and noise can severely reduce the accuracy of these OCR models. In terms of noise, random dust particles can be seen in old flatbed scanners which are not maintained correctly, or if old books are being scanned, we may see folded or yellowed pages, stained pages, etc. All these problems lead to worse output when used for automating of information retrieval / extraction from a pipeline of scanned document images.

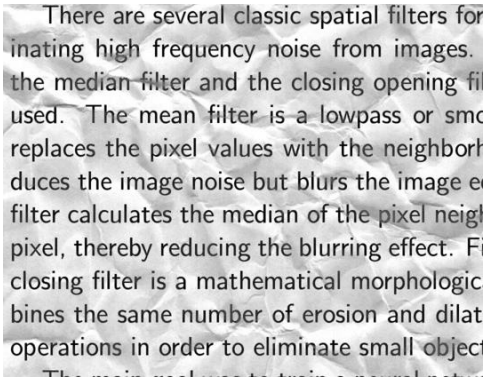
Problem Statement

The problem that is being tackled here is the task of noise removal (denoising) from scanned document images. An autoencoder neural network will be used for this task. It will be trained on the noisy and cleaned versions of the documents and will be tested on previously unseen noisy documents to see how it performs. With additional data, this model can be extended to a larger variety of noise in documents as required.

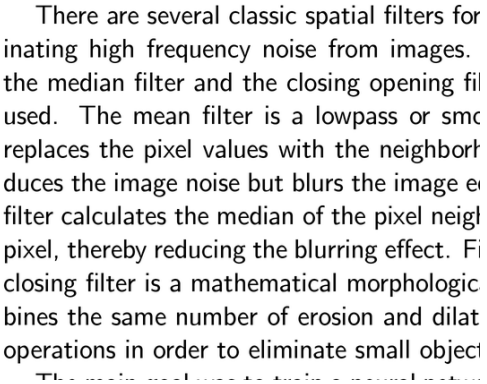
Datasets and Inputs

The dataset is available at the Kaggle Competition link here, and also available at the UCI Machine Learning Repository here.

It contains 144 images each of noisy and cleaned input images – these will be used for training the model. Additionally, 72 test noisy images are given, these will be used to evaluate the model.



Noisy image sample from the training set



Clean image sample from the training set

Solution Statement

The solution to this problem would be to use an Autoencoder network to remove the noise in the image. Autoencoders are used for anomaly detection and noise removal, so it would be suitable for this task.

Benchmark Model

The top submission on the Kaggle competition right now has a score of 0.00416 (RMSE), so this will be used as a benchmark. However due to it being a competition, the exact approach that was used to achieve this score is unknown but considering the problem statement the use of an autoencoder model can be assumed.

Evaluation Metrics

Mean Square Error will be used for evaluation as a loss function. For the generated output files, Root Mean Square Error (RMSE) will be used, as this is being used by the Kaggle competition on submission of results. Each pixel of the output image is compared to the testing set which is used by the competition.

RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Image credits: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Where, y_j is the actual value and $\hat{y}_j$ is the predicted value. Since the errors are squared before they are averaged, the RMSE gives high weightage to larger errors.

Project Design

The following workflow can be implemented for this project:

- Data Preparation
    - Unzipping the dataset into correct folders
    - Reading the images into memory with resizing and normalization of pixel values
- Model Creation & Training
    - Simple autoencoder model to be created – Tensorflow and Keras will be used
    - A few convolution layers, followed by a pooling layer for the encoder part
    - A few convolution layers, followed by an upsampling layer for the decoder part
- Evaluation
    - Predict (denoise) the output image based on the corresponding noisy input image.
    - The autoencoder model will be compiled with the mean square error loss function, since the Kaggle competition calculates RMSE score for obtaining a position in the leaderboard.
    - Mean absolute error will be used as a metric along with Adam optimizer.