**Definition**

*(approximately 1 - 2 pages)*

**Project Overview**

In this section, look to provide a high-level overview of the project in layman’s terms. Questions to ask yourself when writing this section:

* *Has an overview of the project been provided, such as the problem domain, project origin, and related datasets or input data?*
* *Has enough background information been given so that an uninformed reader would understand the problem domain and following problem statement?*

In this project I will develop a proof of concept deep learning solution for self-driving trains. Although solutions are being developed in both the automotive and rail industry for self-driving vehicles, the technology is still not completely mature, and hence it is still not widely adopted. Train delays, and accidents are major problems for train operators, both public and private, that could be addressed with driver assistance or completely autonomous self-driving trains. Although there are existing solutions for driverless trains, the problem in most countries is that it requires dedicated infrastructure, signalling systems, and specific rolling stock, all of which requires significant investment.

**Problem Statement**

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

* *Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?*
* *Have you thoroughly discussed how you will attempt to solve the problem?*
* *Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?*

For this proof of concept I will use a toy train set. The toy train will use image recognition to recognize signals along the train track, and will automatically react to those signals (slowing down, stopping, starting, and accelerating). I will use a raspberry pi mounted to the train to feed images to an amazon AWS GPU instance. A deep neural net will be implemented for image recognition. Training images will be hand labelled. The network will predict if there is a stop sign on the image seen, and also the distance from the stop sign. The raspberry pi will use this information to stop the train, or to start it again if the stop sign disappears.

**Metrics**

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

* *Are the metrics you’ve chosen to measure the performance of your models clearly discussed and defined?*
* *Have you provided reasonable justification for the metrics chosen based on the problem and solution?*

The performance metric for this task will be the accuracy of identifying the presence of a stop table. As this is categorical classification task, the F1-score was the measure I used.

**Analysis**

*(approximately 2 - 4 pages)*

**Data Exploration**

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
* *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*
* *If a dataset is****not****present for this problem, has discussion been made about the input space or input data for your problem?*
* *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

The data for this project has been collected by myself. I have used the raspberry pi’s camera module to take images. There were two categories I needed to collect images for. One category contained a stop sign, the other category did not. The challenge for the data collection was to have images that are similar to each other except for one thing, the presence or absence of the stop sign. I solved this by moving the raspberry pi along a fixed path on a platform, simulating how the train would actually move, while capturing images. The platform was extending out in front of the camera and I could mount the stop sign on it when capturing images. In the other condition I left the platform empty. In this way everything was constant except for the presence or absence of the stop sign.

I have collected 400 images for each category. I used 340 images for training, and 60 images for validation. The raspberry pi camera had a 5 MP resolution camera, but I decided to capture images at 320x160 pixels, as this size would allow for online streaming and processing.

**Exploratory Visualization**

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

* *Have you visualized a relevant characteristic or feature about the dataset or input data?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

Below you can see some example images, covering the entire path that the camera was moving on in both conditions.

Stop:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Non-stop:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Algorithms and Techniques**

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

I used a deep convolutional neural net to solve this problem as this is the best approach known today to do image recognition.

Categorizing images is a difficult task for computers as the input space is generally large. For instance, an image of size 150x150 would be transformed into a vector of size 150x150=22500 for a fully connected neural network. However fully connected neural networks do not take into account the spatial structure of the images, and a better approach is to use convolutional neural networks. What convolutional neural nets do in short is that hidden neurons are not looking at every pixel from the input space, but see only patches of it, for example 3x3 pixels.

In my initial model I have applied 3 convolutional layers, with 3x3 convolutions and 32 output filters applied to each followed by pooling layers of 2x2. This was followed by flattening the last pooling layer followed by a fully connected layer, dropout, and a final single output neuron to perform the binary classification task.

I took advantage of a built in feature of Keras for generating batches of augmented data from a directory. I only had to copy the images for the two categories into different directories, and Keras took care of data augmentation and applying labels for the binary classification.

Data augmentation is useful when one has only a small sample of images. Augmentation consists of small transformations applied to the image, which helps prevent overfitting, as we have a larger training set to learn from. The specific transformations I have applied to the images were: rescaling, rotation, shifting width and height, shearing, and zooming. Below you can see some examples:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

I had to fine tune with how much to transform, so that the stop sign remains always on the image. If too much width/height shift, or zoom is applied, than it can happen that the stop sign disappears from the picture.

**Benchmark**

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* *Has some result or value been provided that acts as a benchmark for measuring performance?*
* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

For this task the benchmark was accuracy, and I aimed for an accuracy of at least 95%, as I have found this level of performance to be achievable even in multi category classification for traffic signs (e.g. http://yann.lecun.com/exdb/publis/pdf/sermanet-ijcnn-11.pdf).

**Methodology**

*(approximately 3 - 5 pages)*

**Data Preprocessing**

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the****Data Exploration****section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

As mentioned earlier, data preprocessing consisted of reducing the resolution of the images (such that it is manageable to train a neural net in reasonable time even on a simple laptop with CPU, and to allow for online streaming and processing in the future), and image transformations. Images were captured with size 320x240 pixels. These were further scaled to 150x150 pixels by Keras. Transformation of images were done with the following parameters:

* rescale=1./255: as the images taken by raspberry pi’s camera come with RGB coefficients in the range of 0-255 I had to normalize the values to span from 0 to 1., which was achieved by this scaling
* rotation\_range=20: images were rotated randomly by 0-20 degrees
* width\_shift\_range=0.01: range in which image was randomly translated vertically
* height\_shift\_range=0.1: range in which image was randomly translated horizontally
* shear\_range=0.05: range in which shearing transformations were applied randomly
* zoom\_range=0.1: range in which image was zoomed at randomly
* fill\_mode='nearest': this was the method with which newly introduced pixels were filled out

**Implementation**

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

For the implementation I have chosen Keras. Keras is a neural network library for Theano and TensorFlow written in Python. As I have already mentioned, a convolutional neural net architecture was applied for the task. The network consisted of an input layer, 3 convolutional layers, a fully connected layer, and an output layer. The convolutional layers used 3x3 convolutions and 32 output filters followed by pooling layers of 2x2. For the activation functions I have chosen rectified linear units, except for the final output neuron which was sigmoid. After the fully connected layer a dropout of 0.5 was applied (this helps to prevent overfitting). For the loss function I have used logloss (binary crossentropy). The initial optimizer was rmsprop.

**Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

As even the first implementation gave an accuracy of >98% on the validation set, I only tried minor variations, including a change in the optimizer (change rmsprop to adam), and changing the receptive field of the convolutional layers from 3x3 to 2x2. Below are the initial results followed by the results with the modified parameters:

**Results**

*(approximately 2 - 3 pages)*

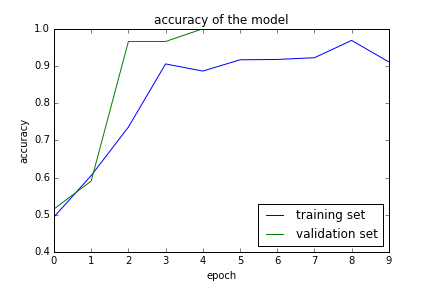
**Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called *sensitivity analysis*). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

I have used a separate validation set to test the robustness of the model. The final model reached 100% accuracy on the validation set. Such a high level of accuracy is very important given the nature of the application (train control).

As I have applied both dropout and data augmentation, I believe the model can be trusted, at least as long as the train is moving along the same path where the training data was gathered. It would be good to test how robust the model is if we change the path, and introduce a very different environment.



**Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

The result obtained with the final model was higher than I initially expected. I thought that an accuracy of around 95% would be impressive enough for this particular task. In the specific context it seems to have solved the problem.

**Conclusion**

*(approximately 1 - 2 pages)*

**Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

In this project I have provided a solution for an aspect of automatic train control using image recognition. Image recognition used to be a difficult task historically, however for the last few years we have discovered efficient methods to approach these kinds of problems. Deep multi-layer neural networks are capable of building up a hierarchy of abstractions that makes it possible to identify complex inputs (i.e. images), and in this project I have utilized such an approach.

There were two major areas for the project. The first was data collection, the second was model building. Before I could actually build an accurate model, it was important to have a sufficient amount of data. Given that I was not working with an externally provided dataset a major difficulty for this project was to figure out an efficient method with which I could collect enough data. The reason why it is good to collect a large amount of data is that it helps to prevent overfitting and improves generalization. I think I have managed to collect a sufficient amount of data for this particular problem, and with data augmentation I had introduced quite a bit of variation, which can helps in generalization.

After this, the actual model building was not that difficult. Although there are a large number hyperparameters I could experiment with (such as the number and type of layers, number of neurons per layers, the type of activation functions, regularization methods, loss functions, error metrics, optimizers), I could start from good architectures that were already discovered and published by others, and build from there. For this particular problem, although I had already extensive experience with using neural nets for demand forecasting, I had no prior experience with using neural nets for image recognition. It was amazing to see how efficient this method is, and how fast I can set up an architecture that is performing really well on the task.

Although the final method fits my expectation for the problem, the specific model would need additional testing if it were to be used in an environment that is much different than the one in which I have developed it. The bottleneck here was the difficulty around data collection, however additional data could be gathered in case the model does not generalize well enough in largely different environments.

**Improvement**

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting your report, ask yourself…

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly Analysis and Methodology) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?