## Overview

The system will begin by processing the images to prepare them for analysis. Once the images are ready, the deep learning network will begin analysing them.  
The system can essentially be separated into 2 major sections. The first section will be a deep belief network aimed at identifying the damaged areas and classifying the severity of the damages. The classifications from the first section and the original image data will then be passed into the second section consisting of convolutional layers and pooling layers with the aim of producing the final cost estimate.

#### Image Pre-Processing

#### Deep Belief Network

The deep learning system will start with a deep belief network aimed at identifying the damaged areas of an image. A deep belief network is ideal for this as it consists of Restricted Boltzmann Machines (RBM) which are effective at identifying the most important aspects of data first. We would probably have to play around with the exact number of layers that is most appropriate, however, we expect that it would be between 5 and 10.

A RBM is a neural network which runs as normal but is then reversed. This means that all of the weights now work in the opposite direction. The output of the original network is then taken and passed through the reversed network. The reversed network should produce the original input back as output. By doing this, no expected output is required to train a RBM. As mentioned before, these RBMs tend to identify the most important aspects of the input data first.

RBMs can also be used to reduce the number of neurons required during processing. Which will increase training time and processing time.

A deep belief network consists of several RBMs in sequence. This is helpful because each RBM can be trained individually, one after the other, instead of training all of them at the same time. This also makes training much shorter.

#### Softmax Classification

Once the deep belief network has identified the most important aspects of the input data, the network will make use of the softmax activation function to classify the damages and rate the severity of the damages.

This would be best achieved using at least two layers. The first layer would classify the damages. The second layer would use the original input (the output of the deep belief network), via deep residual techniques, as well as the classification from the first layer to classify the severity of the damages. These classifications will provide input for the rest of the network and should increase the accuracy of the system in general.

#### Deep Residual Learning

One of the issues often faced when working with deep neural networks is that deeper layer neurons tend to train faster than shallow layer neurons. As a result, the shallow layers output inaccurate data (because they are still training) and the deeper layer neurons are trained on this inaccurate data. The deep layers learn to ignore the input data and simply output average results to produce a medium error on average. By the time the shallow layers are trained and producing accurate data, the damage is already done and is very difficult to correct.

Deep residual learning helps avoid this by adding extra connections from the initial input directly to deeper layer neurons. This way, the deeper layers can rely on some accurate input while the shallow layers are still training.

Seeing as RBMs are trained one layer at a time, they wouldn’t have this problem and therefore wouldn’t require deep residual learning. The convolution layers, pooling, etc. (to be discussed below) will require deep residual learning. As mentioned before, the softmax classification layers would also make use of deep residual learning.

We would apply deep residual learning primarily to the second half of the neural network (after classification). This will allow us to train the convolution and pooling layers using both the initial image data as well as the classifications obtained. Initially, we will start with quite a lot of residual connections and gradually remove connections as the system is trained. We will remove connections from the shallower layers first as these layers are less affected by the issue at hand. The residual connections would also have constant weights as the network already has a lot of weights to train. Plus there isn’t much point training weights which will eventually be removed.

#### Convolutional Layers

#### ReLU

#### Pooling Layers

#### Flattenning

#### Dropout

Dropout is a technique used to reduce the training time required for deep neural networks.

The idea is to select random neurons each iteration which will be ignored. This reduces the number of neurons which have to be processed each iteration, thereby making the training process faster.

Dropout also has an added bonus where the network is trained to solve the problem using different neurons each iteration. As a result, it learns to solve the problem in different ways and becomes much more robust.

Dropout is not applied during standard running of the neural network, however. It is only applied during training. Because of this, the network is actually run with more neurons than it was trained with and thus the output needs to be scaled down in proportion to the dropout.

We would start by using a 20% dropout rate as this is fairly standard, but we would experiment with different values to get the best results.