**Main Idea**

The main idea of the paper is to understand the reason behind convolutional neural networks performing so well and how they can be improved. The authors introduce a novel visualization technique that gives more information about the function of intermediate layers, the visualization helps to understand what features of the data excited the activation maps. These visualization techniques can be used for debugging and to find better models. In addition to that, the authors have conducted various experiments to identify some characteristics of deep networks, they developed a new model which outperformed Krizhevsky’s model (a good model) on ImageNet, they were able to develop that model by using insights from the visualizations.

**Visualization with Deconvnet**

The visualization technique proposed is called “Visualization with Deconvnet”. Deconvolution can be thought of as the inverse of the convolution neural network, each step in the forward pass is reversed. The aim of the deconvolution is to understand what input features excited the given activations the most at each layer in the conventional convnet. We pass the activation maps of the layer to the deconvolution network. The deconvnet will try to undo all the changes that were made in the forward pass till that layer, it will map the activities back to the input image thereby showing what properties of the input caused the activation in the feature maps.

The below image is taken from the paper and will be used to explain the deconvnet in detail.

Diagram

Description automatically generated

The deconvnet is connected to each layer of the conventional convnet. To understand the features learnt in a layer, we pass the activation from that layer to the deconvnet, the rest of the activations are set to zero. There are mainly three operations that need to be performed:

**Unpooling:** This can be thought of as undoing the work performed by a max pooling layer. It is not possible to get an exact reconstruction by unpooling but we can get a good approximate. As shown in the image, there are switches connecting from max pool layer in the convnet to the unpooling layer in the deconvnet. The purpose of the switches is to maintain the state where the maxima of the pooling regions occurred. During unpooling these switches are used to place the reconstructions from the layer above into locations such that the same state is preserved as stored by the switches. In the image above, the unpooled output has maximums in the same location as the original input however the magnitude is different.

**Rectification:** The goal here is to undo the effects of the non-linearity which in this case is relu. The effects of relu cannot be undone so we use another relu to approximate the rectification.

**Filtering:** This can be thought of as deconvolution. We will use the filters which were learnt by the convnet, but instead of directly using them, we will transpose them and then apply to the outputs from the layer above (rectified units).

The above three steps are used to reconstruct the layer beneath that generated the activations (the activations which we have undone). The same process is repeated until we reach the input pixel space.

**Model**

The authors trained their model on the ImageNet data, they created new data by cropping/ flipping original images to expand training size. A summary of their model: 7 layers, Layers 1:5 convolutional layers including maxpooling and activation functions, Layer 6:7 fully connected layers, at the end the softmax layer.

**Feature Visualization**

The feature maps from the different layers are projected back to the input space. For each layer, it can be seen what features of the image the layer has detected, layer 2 was able to identify corners, layer 3 captures textures etc.

**Feature Evolution**

The authors demonstrate the feature evolution during training. It is seen that the layers learn better as the epochs increase and the lower layers converge faster compared to the deep layers. The deep layers require more epochs for good level of convergence, the features learnt by the deep layers are better visible with higher number of epochs.

**Feature Invariance**

The authors also depicted the impact of image translation, rotation and scaling. The result was that the small transformations impacted the first layer significantly, but the last layer was relatively unaffected and there was no change in the classification. The output is not invariant to rotation unless the image has rotational symmetry.

**Using Feature Visualisation for debugging**

The authors demonstrate that the visualizations from the deconvnet are helpful to identify problems in the architecture. They detected two problems in the existing convnet architecture (Krizhevsky’s architecture) and slightly modified the architecture (changed the dimensions of the filter and the strides). The modified architecture yielded better results and outperformed the prior work. With the modified architecture, the feature representations conveyed more information.

**Occlusion Sensitivity**

The authors also depicted that the model is truly identifying the location of the object and not just predicting based on the surrounding context. To prove this, they placed a patch at different positions in the training images and observed the results. The result was that when the patch was placed on the face (the criteria for classification), the classification error was high which proved that the model was able to identify the true location and that model predicts using the correct information. In addition to that, they showed that when the grey patch covered the positions in the visualizations (deconvnet visualizations), the feature map activity also reduced. This proved that the visualizations contain those features that truly stimulate the feature maps.

**Correspondence Analysis**

The authors also covered correspondence analysis. This basically means that the deep learning models are not able to explicitly establish relation between the same object part in different images. A simpler explanation for correspondence analysis is: if I change a specific part of an image (remove the nose from a dog image), the feature representations change in a consistent way even if I do this for any dog images. They did an experiment where they cropped out a specific body part from each of the dog images and then calculated the difference between the original and the cropped image for each of the sample images. They compared the consistency of this difference across the sample images, and it was found to give a low value. The low value indicated greater consistency between the same part in the different images. This shows that the deep models are implicitly computing the correspondence.

**Experiments**

The authors ran several experiments. The used the ImageNet 2012 data and compared the performance of their modified model to prior work. The model outperformed the previous work. They also did an ablation study in which they removed some layers of the model and checked the classification performance. Changing the size of the fully connected layers did not have much effect but increasing the size of the middle convolution layers improved the performance. The conclusion was that having a minimum depth to the model is necessary rather than having a specific section, because removing the fully connected layers or just removing the two middle convolutional layers did not affect the classification by a lot.

They generalized their model by training it on different datasets, they only modified the softmax layer of the model and were able to achieve good levels of accuracy. They outperformed the prior work on Cal-101 and Caltech-256 data sets. But the model that they trained from scratch on these datasets performed porrly. Their model did not perform well on the PASCAL data, but it was primarily due to the difference in the images. The PASCAL dataset contained multiple objects, but the author’s model was trained on ImageNet data which had prediction for a single image.

**Feature analysis**

The authors explored how differentiated the features in each of the model of their convnet are. They did so by varying the number of layers from the pretrained imagenet model and placed a SVM/softmax classifier. It was seen that the best results (accuracy) were obtained with deeper models, this also makes sense as when the model gets deeper, it is able to learn more complex features which help in better classification.