Coordinate based discriminative image model (Coordinet)

Jeremy Gillen

Introduction

A recent paper (Anokhin et al., 2020) has shown that it is possible to create a powerful generative model for images, comparable to state of the art models, without using convolutions. The model takes a random latent vector, and a pixel coordinate, and generates a single pixel from the image. Repeating for each pixel, an impressively realistic image is generated. This suggests that the model, using just pixel coordinates, contains similar inductive biases to those encoded by convolution, or contains different inductive biases that are just as useful for generative modelling. I propose investigating whether similar inductive biases can be embedded in a discriminative model that classifies images without using any convolutional layers. My proposed architecture is made up of an encoder and decoder network, with the encoder network mapping each (single pixel, coordinate) to a latent vector, then summing over all latent vectors into a single vector representing the image, and passed to the decoder network that maps the latent vector to a softmax classification output.

The discriminative model will be compared with simple fully connected and convolutional architectures, in the hope that it will generalize better than fully connected networks, and be comparable (but probably a little worse) to the CNNs. Architectural variations will be investigated, including the use of multiplicative conditioning (ModFC layers) and different positional encodings, as found in Anokhin et al. (2020).

The next step in this investigation will be to make a comparison between the inductive biases of convolutional architectures, fully connected architectures, and the coordinate-FCN architecture. This will be conducted by comparing the test error of each of these architectures, for various small dataset sizes on a variety of image datasets. Further investigation is presented into the precise differences between the inductive biases of convolutional layers and coordinate-FCN architecture, by comparing robustness to various input transformations on toy datasets (rotation and scaling).

Importance:

As in (Anokhin et al., 2020), processing pixels individually with their coordinates has the potential to be useful for encoding different types of prior information. With various positional encodings, images with a wide variety of topological shapes can be easily and accurately represented, including spherical or cylindrical images. Another application is encoding prior information that an image is warped in some way, like in the case of images taken with fish-eye lenses. Other potential benefits of this architectures include it's highly parallel nature, and ability to easily process different image resolutions, and even process different resolutions within the same image.

The hypothesis being tested

Where does the inductive bias of convolutional networks come from? Why does it work? Two explanations come to mind:

1. Convolutions force the network to learn a hierarchy of position invariant "concepts". Since objects in the real world can usually be easily represented in this way, this prior information

helps the network to successfully generalise. This explanation appears to be the most common way of explaining the inner workings of convolutional networks (Olah et al., 2017).
2. The usefulness of convolutions comes from their ability to encode prior information about the relative locations of pixels. This idea would explain why (Anokhin et al., 2020) was able to create a generative model of comparable quality to convolutional networks, while only generating one pixel at a time and without using a hierarchical structure.

Another explanation for the results of (Anokhin et al., 2020) is that their network *is* actually doing something like 1, but in a way that is more difficult to see.

Related works

A 2018 paper (Liu et al., 2018), which partially inspired (Anokhin et al., 2020), investigated adding coordinates as extra channels to the initial convolution operations. This paper shows advantages of this approach for generalisation and model size.

Techniques

Modulated Fully Connected layers (ModFC) are one of the main features of the architecture used in Anokhin et al. (2020). In each layer, the weights leading to each output neuron is multiplied by a different modulating value. Then the weights are normalised. The modulating values are the output of a different fully connected network, whose input in the paper was the latent representation of the image to be generated. In the Coordinet architecture, the entire emage is compressed into a latent representation that is used in the same way. The purpose of these layers is to allow the network to adjust the computations being done on each pixel input, depending on the overall structure of the image. They may be thought of as a simple kind of attention mechanism, which allow sections of the network to specialise on different tasks, and route information through appropriate neurons when they are needed.

My implementation of the ModFC layer differs from the original. The original implementation multiplied the weight matrix directly, which requires duplicating the matrix for every pixel in the image. My implementation simply multiplies the output of the matrix multiplication pointwise with the modulation vector, before a layer normalisation. This allows the same functionality with a simpler and more readable implementation.

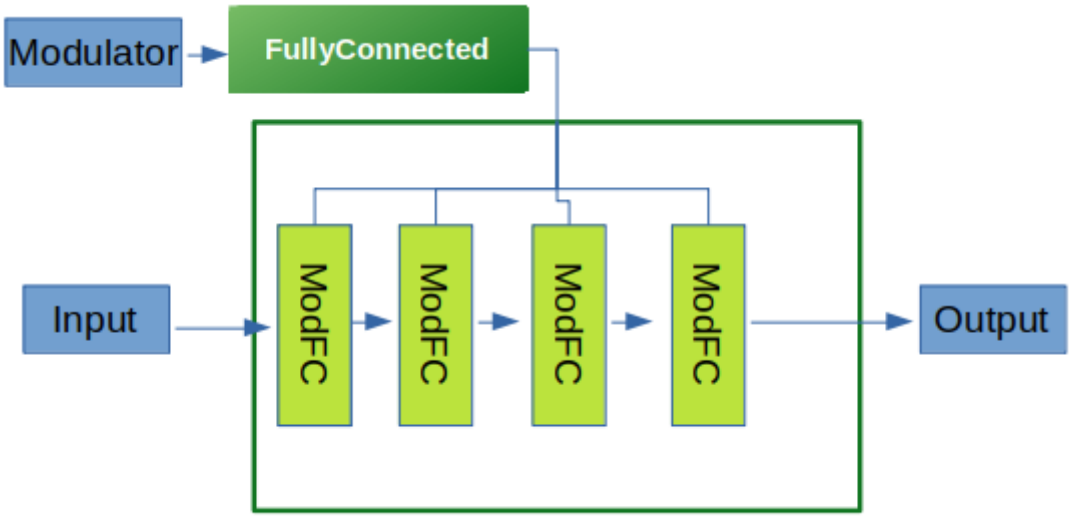
The positional encoding used to encode the position of each pixel is an important part of the architecture. The simplest idea for encoding the pixel position is to simply give the network the pixel indicies, maybe normalised to the range [-1,1]. This works, but can be improved upon using vectors to encode each index. Each element of the vector is calculated using the equation $\sin(ci)$, where i is the (normalised) index and c is a constant that depends on the position in the positional encoding vector. This method means that different positions in the vector indicate the intended position at a different level of precision. Earlier locations give an approximate location, and can be combined with later layers which indicate the precise location.

Summation of vectors in machine learning has a history of being used to combine pieces of information. Word embeddings can be added or subtracted to get meaningful results (Allen & Hospedales 2019), and multiheaded attention mechanisms use a weighted addition to combine information from multiple locations (Vaswani et al., 2017).

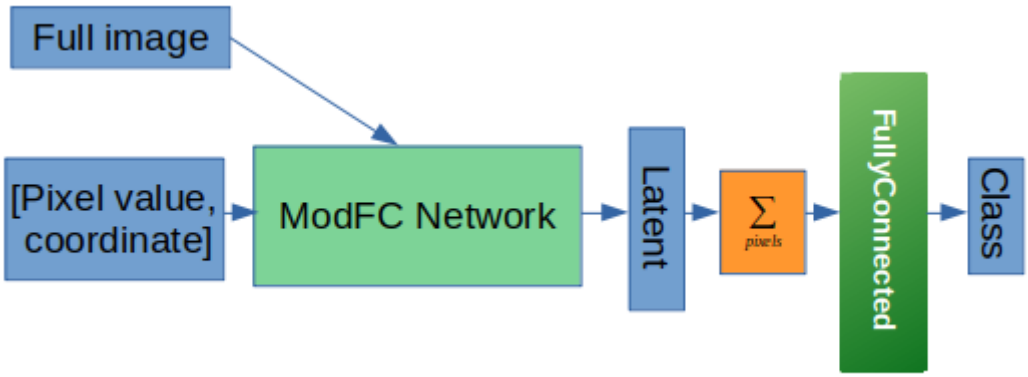
Architecture

Coordinet2

The following diagram shows the ModFC Network architecture, which modulates fully connected layers with a second input signal. The output of each ModFC layer is multiplied by this modulator.



The next diagram shows the full architecture of the Coordinet, where the left half of the image is executed once for every pixel in the image.



The goal in constructing this model was to approximate the same flow of information as used in the generative model of Anokhin et al. (2020). In that model, the network is modulated by the random latent vector used to generate the image. The Coordinet architecture takes the full image and uses that in the place of the random latent vector.

Fully Connected

The fully connected model had 9 hidden layers, each of size 200. Residual connections were used across every layer.

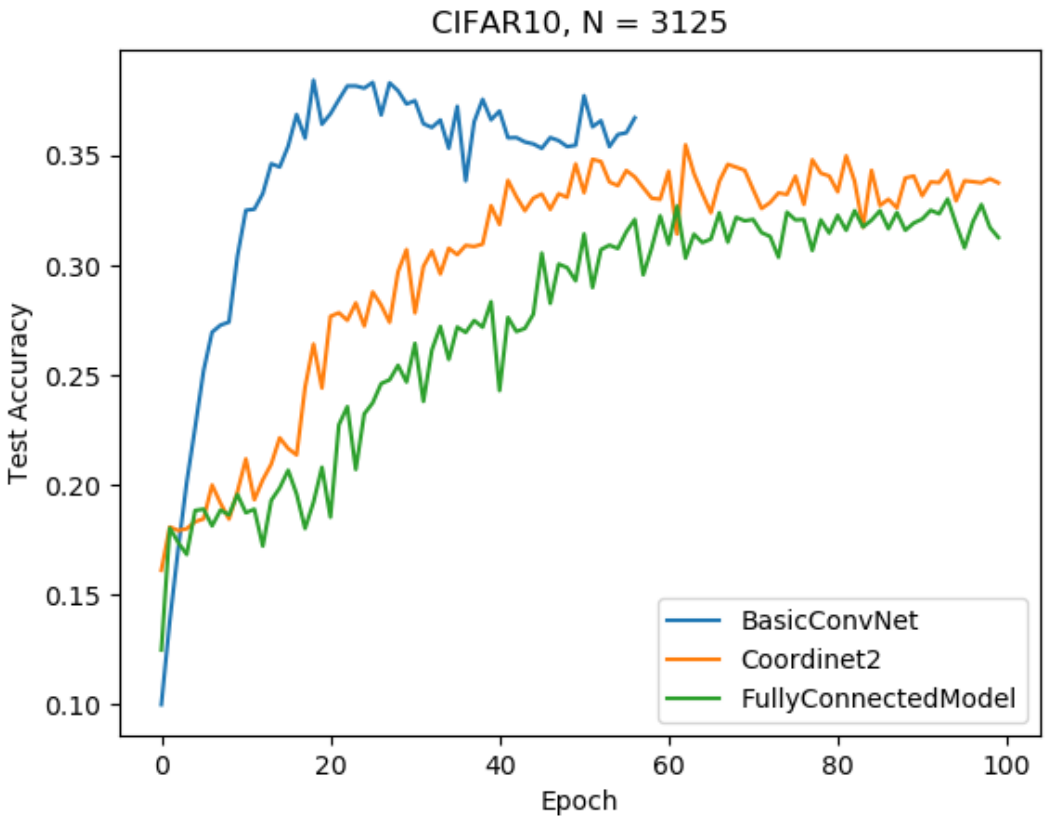
Simple Convolutional

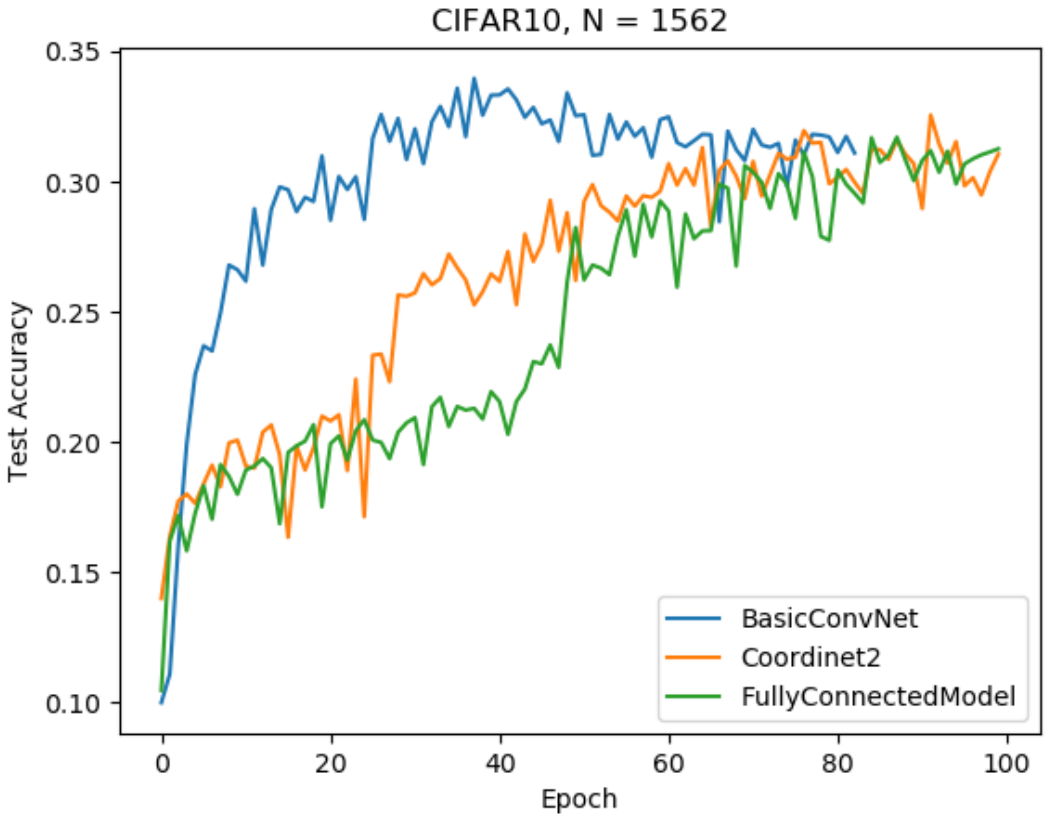
The simple convolutional model consisted of 6 convolutional layers, of increasing width and a kernel size of 5. No pooling layers were used, and a single fully connected layer was used for the output.

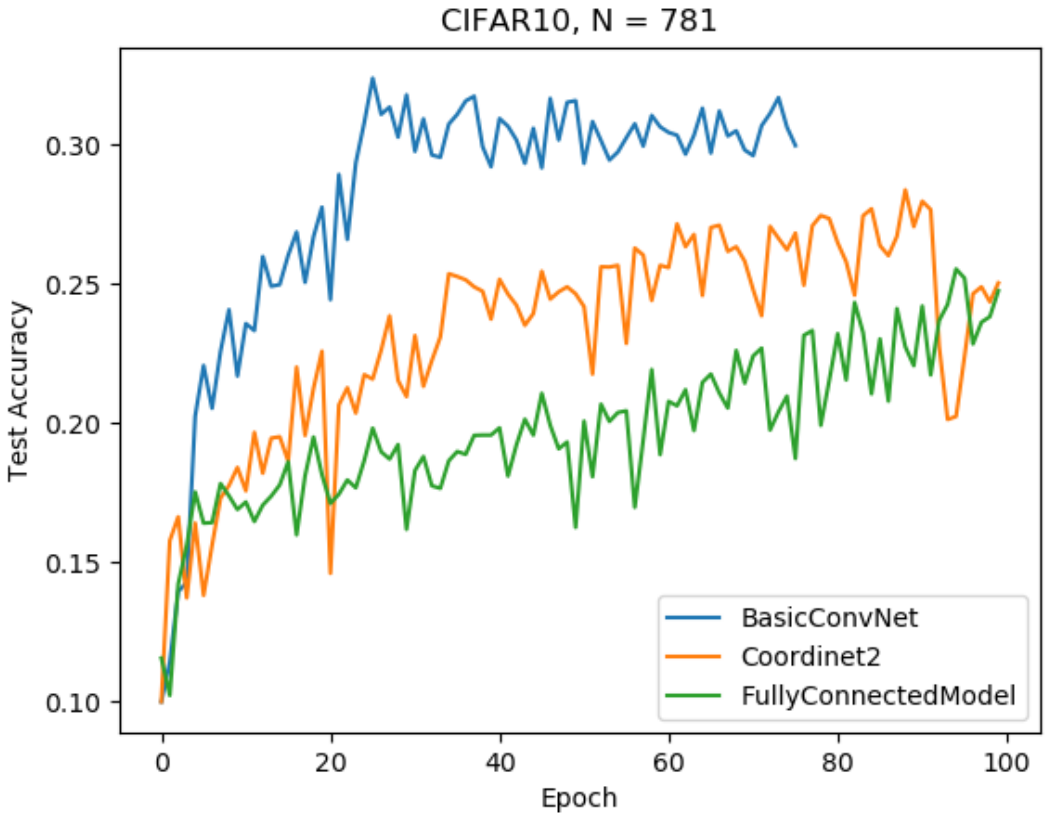
Experiments and results

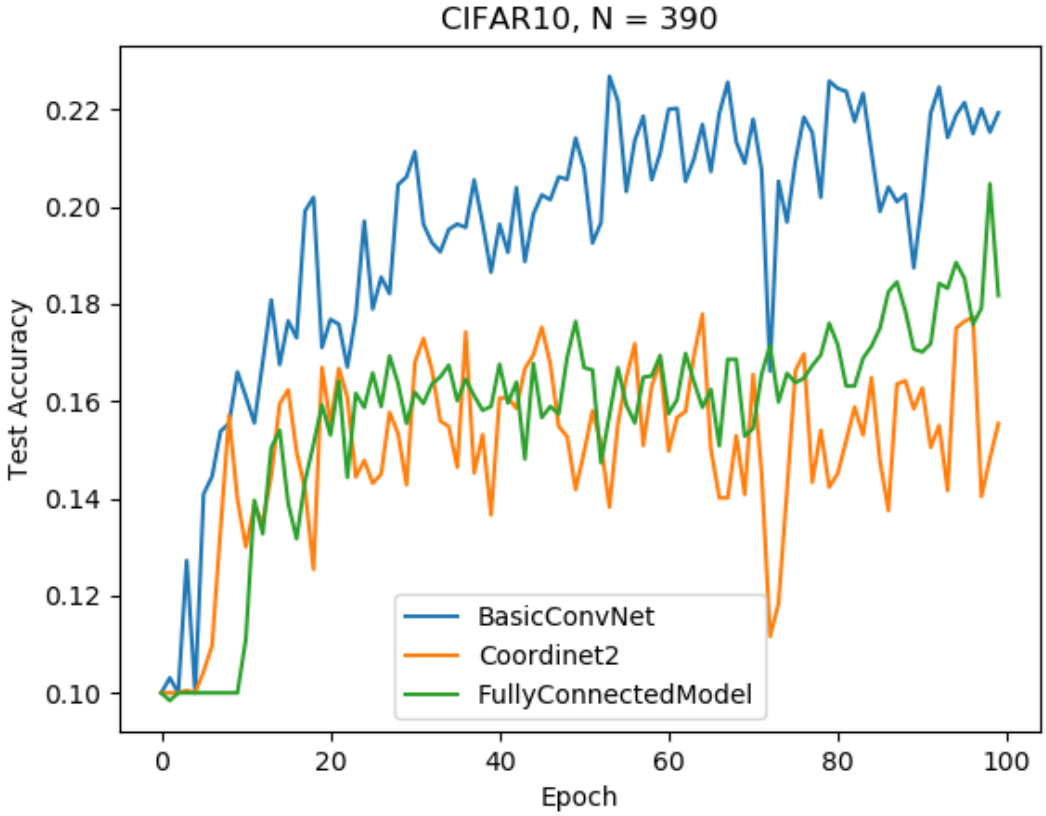
The first experiment trained each of three models to the same loss (0.1), on a subset of the training data of size N. This ensures that each model performs the same on the same set of training data, so differences in their test accuracy should be entirely due to the inductive biases of the model (and random chance from initialisation). Small values of N were selected to amplify differences in the inductive biases, since with large enough datasets, information learned from the data may overwhelm prior information built into the model.

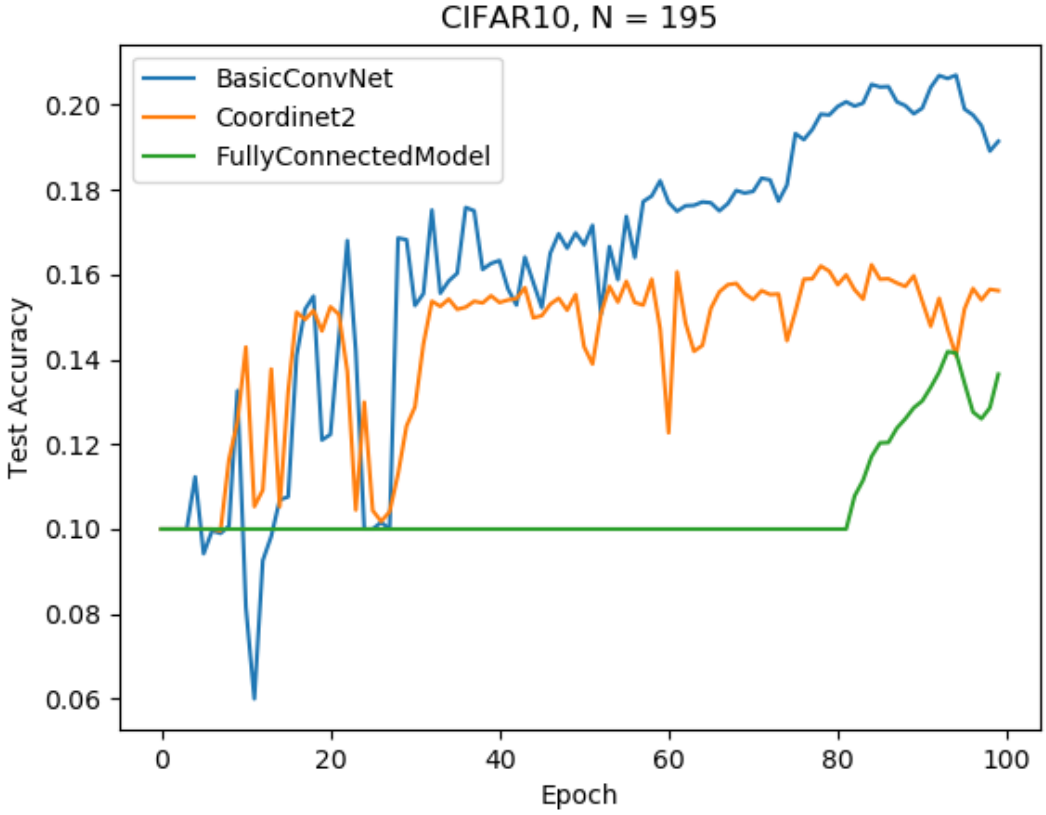
The following graphs show the test accuracy of each of the three models during training. The graphs show that the Coordinet architecture usually outperformed the fully connected architecture, and the convolutional architecture always outperformed both by a large margin. As can be seen in the graphs, the training process is sufficiently noisy that it is difficult to tell how significant these results are.

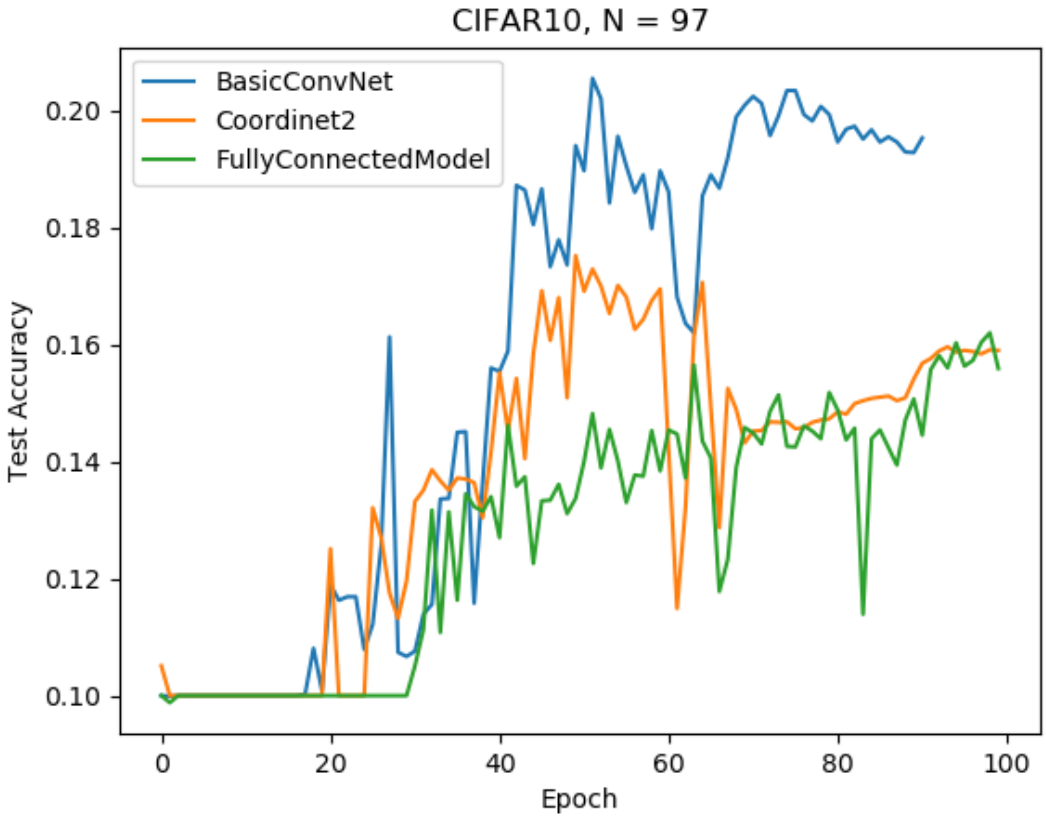



CIFAR10, N = 1562
Test Accuracy
Epoch
0.35
0.30
0.25
0.20
0.15
0.10
0
20
40
60
80
100
BasicConvNet
Coordinet2
FullyConnectedModel

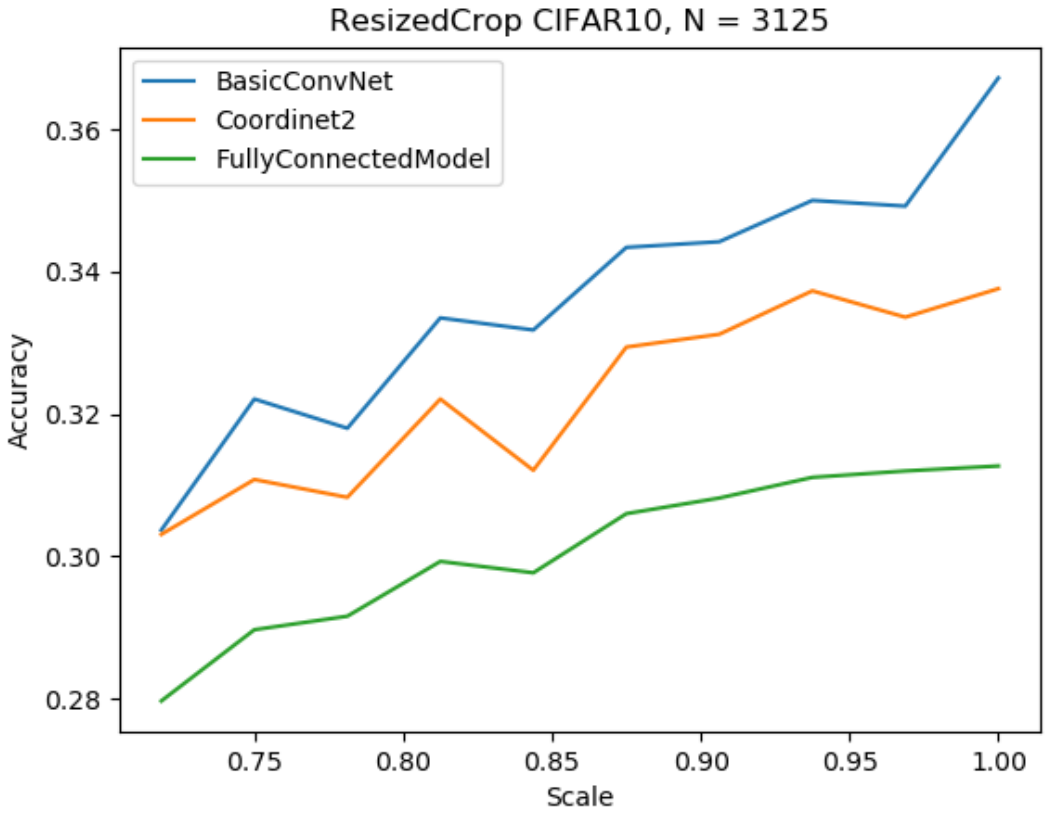

CIFAR10, N = 781
Test Accuracy
Epoch
0.30
0.25
0.20
0.15
0.10
0
20
40
60
80
100
BasicConvNet
Coordinet2
FullyConnectedModel

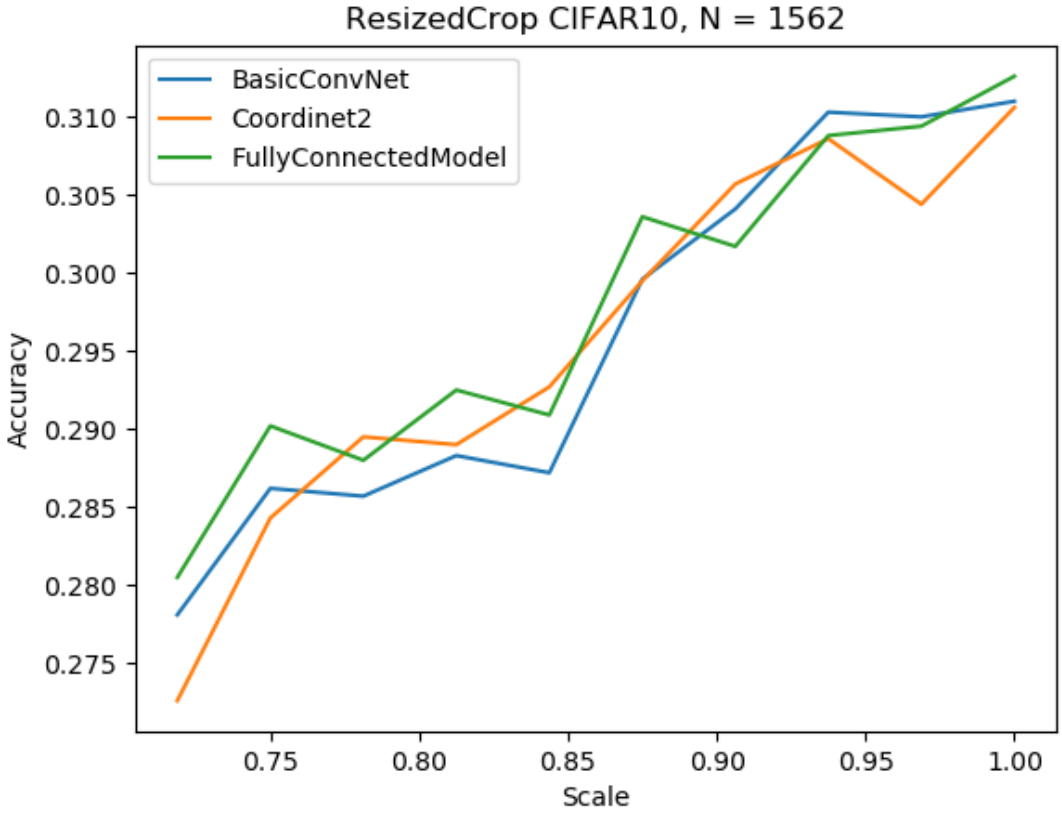

CIFAR10, N = 390
Test Accuracy
Epoch
BasicConvNet
Coordinet2
FullyConnectedModel
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0
20
40
60
80
100

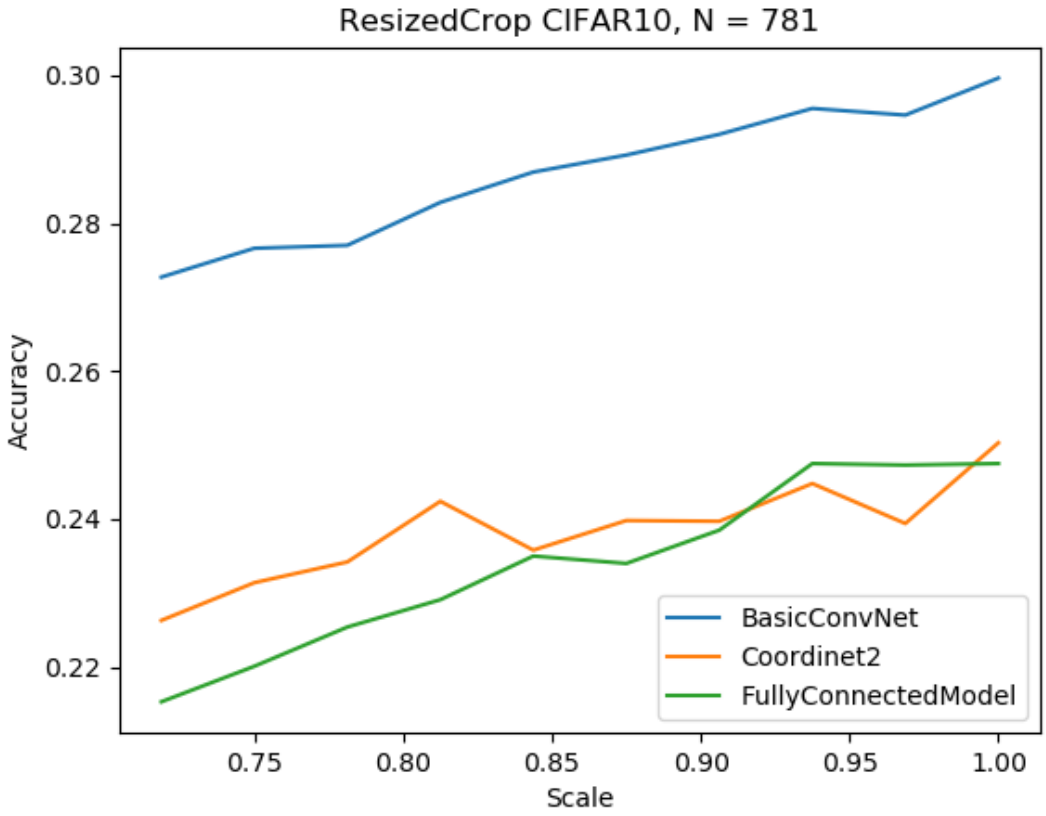

CIFAR10, N = 195
Test Accuracy
Epoch
BasicConvNet
Coordinet2
FullyConnectedModel
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0
20
40
60
80
100

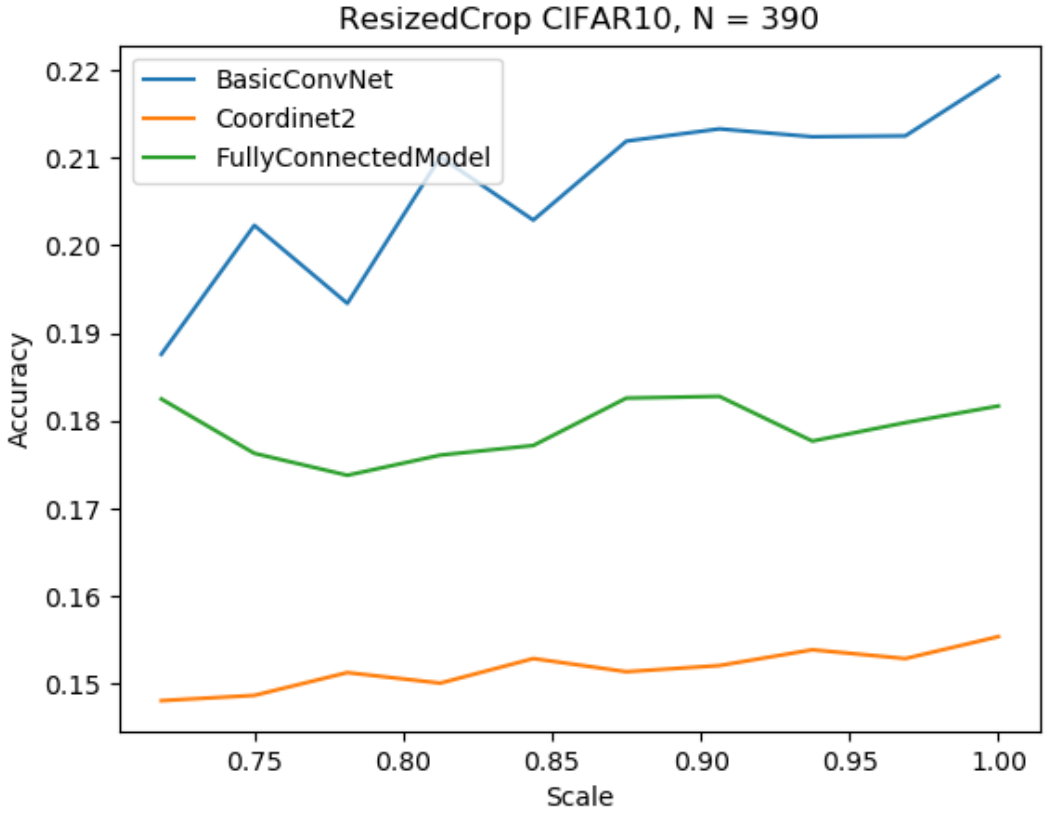


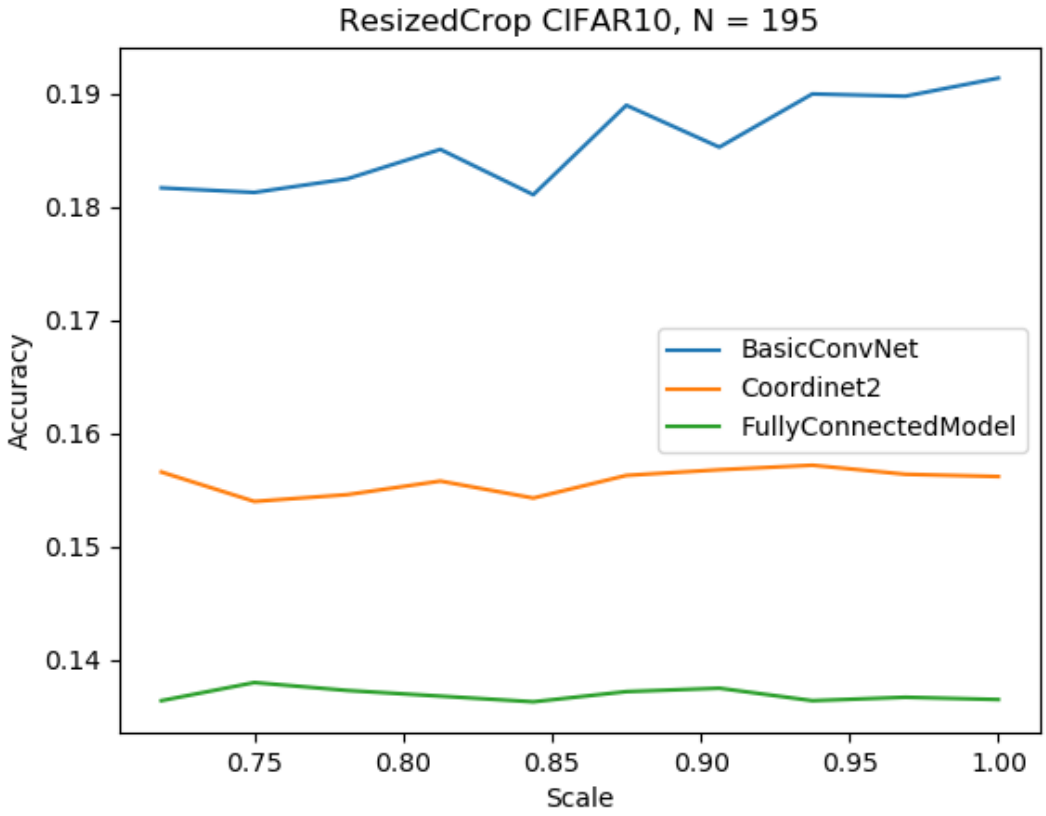
The next experiments applies two types of distributional shift to the test set, to compare the relative generalisation ability of the three models. The first transformation applied involves cropping the image and scaling up the cropped area to the original image size. Each graph shows the model performance across different crop sizes.

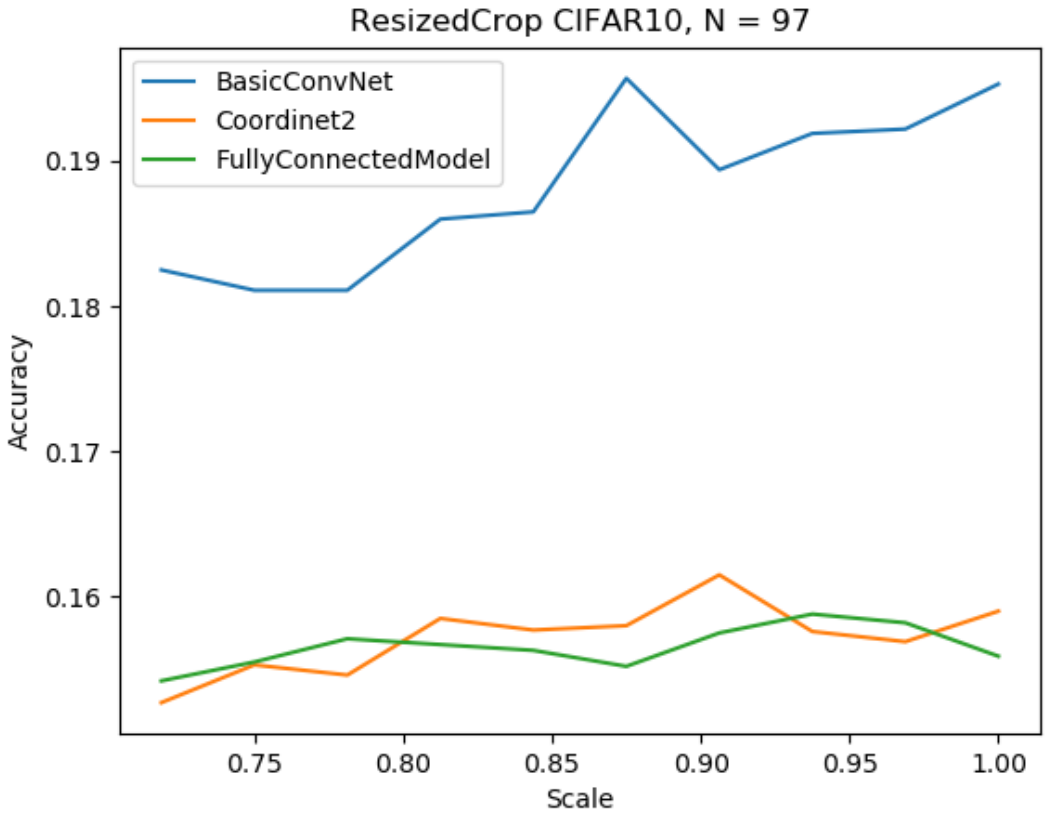



ResizedCrop CIFAR10, N = 1562
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.310
0.305
0.300
0.295
0.290
0.285
0.280
0.275
0.75
0.80
0.85
0.90
0.95
1.00
Scale

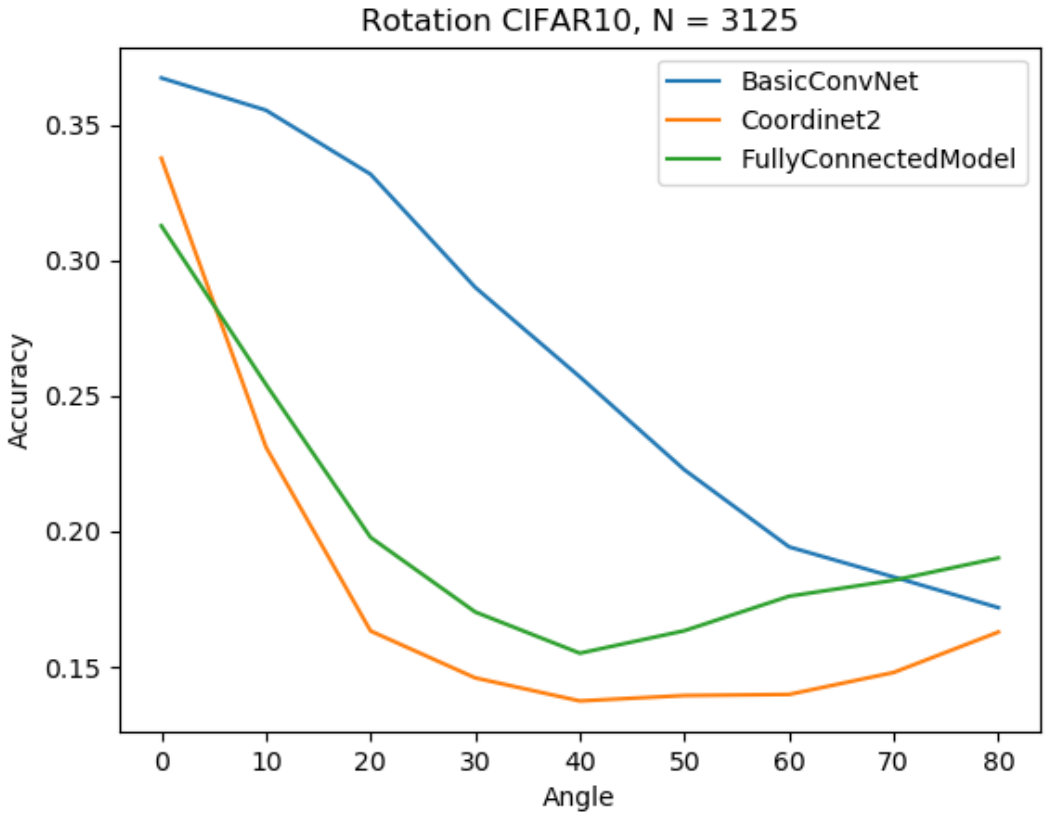

ResizedCrop CIFAR10, N = 781
Accuracy
0.30
0.28
0.26
0.24
0.22
BasicConvNet
Coordinet2
FullyConnectedModel
0.75
0.80
0.85
0.90
0.95
1.00
Scale

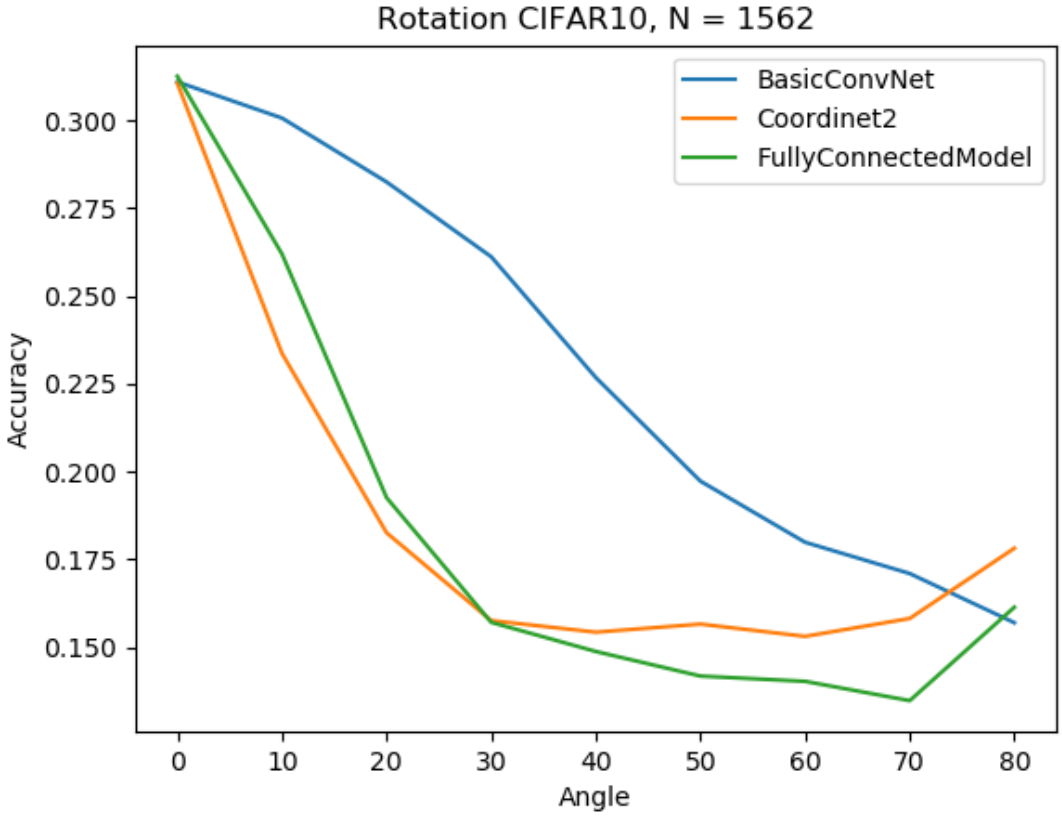

ResizedCrop CIFAR10, N = 390
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.22
0.21
0.20
0.19
0.18
0.17
0.16
0.15
0.75
0.80
0.85
0.90
0.95
1.00
Scale

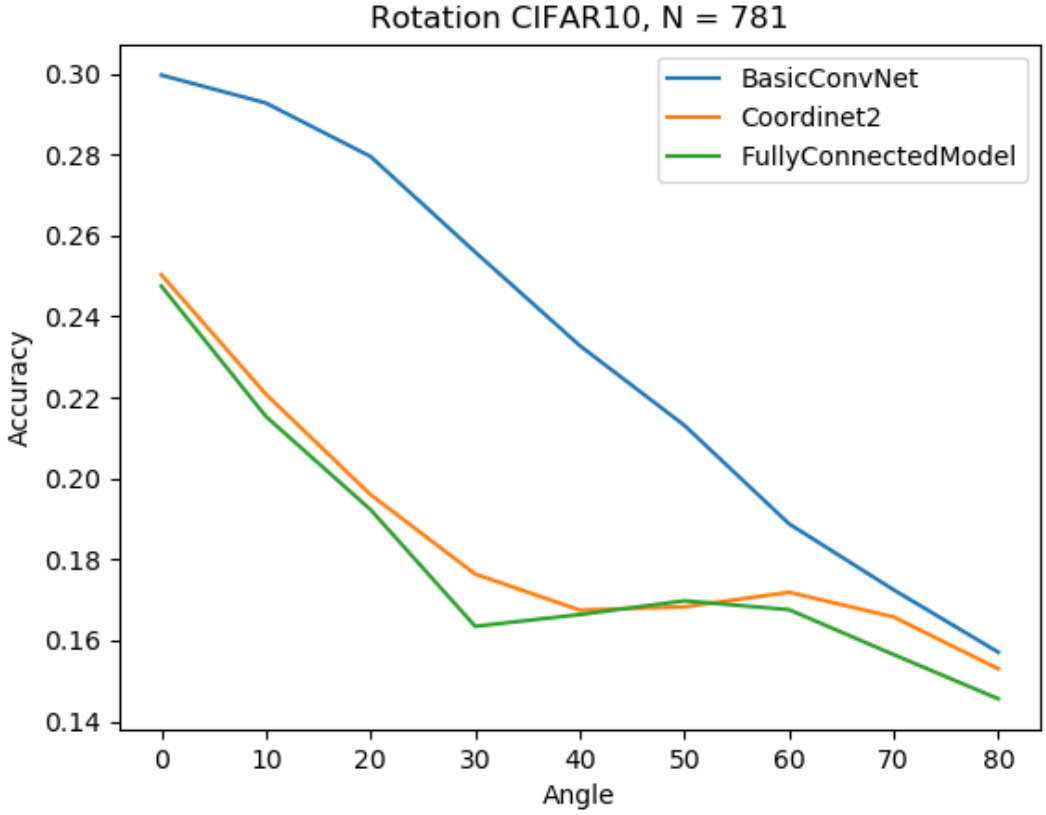

ResizedCrop CIFAR10, N = 195
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.19
0.18
0.17
0.16
0.15
0.14
0.75
0.80
0.85
0.90
0.95
1.00
Scale

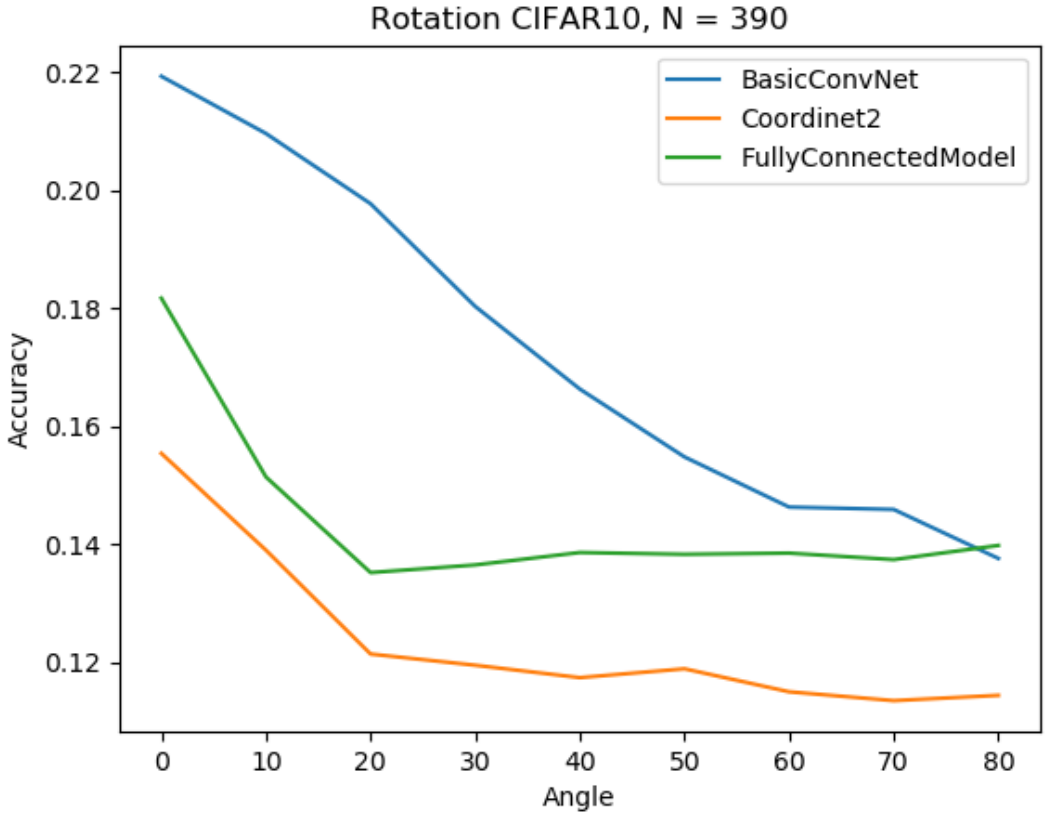


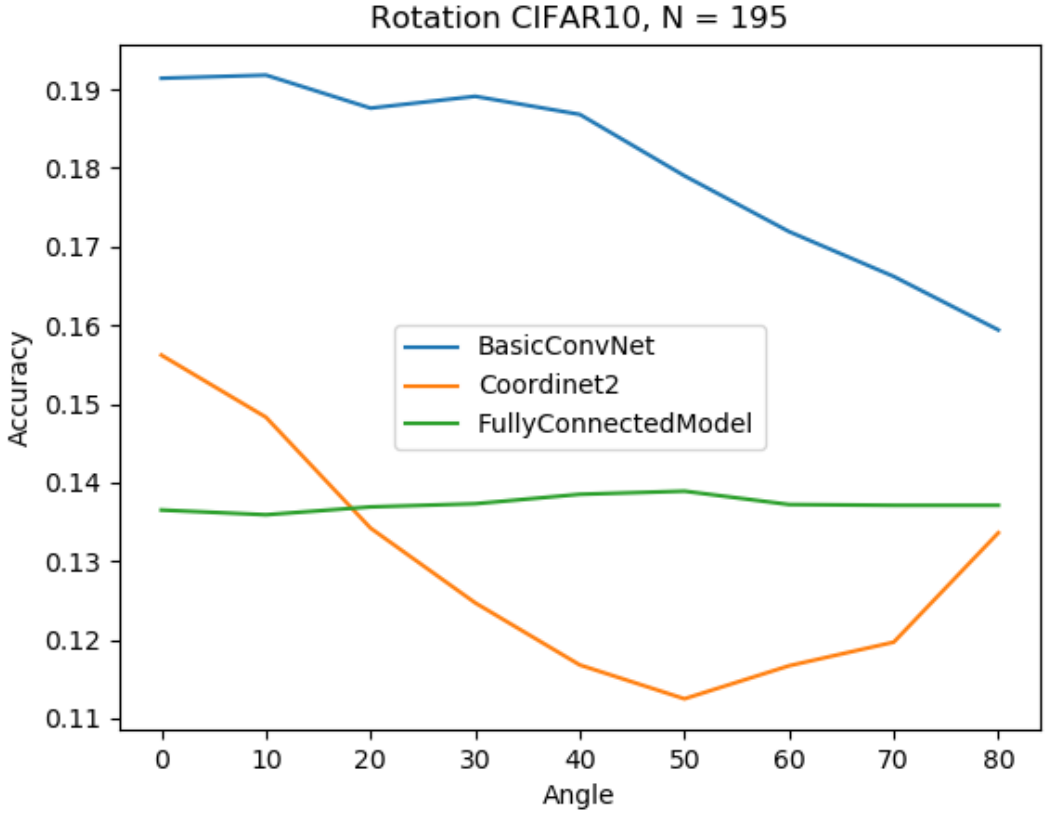
The following graphs show the same experiment except that the test images are rotated instead of scaled. The graphs show good generalisation performance by the convolutional model, which mostly outperformed the other two models by a large margin. On this test, the Coordinet architecture usually generalised the worst of the three.

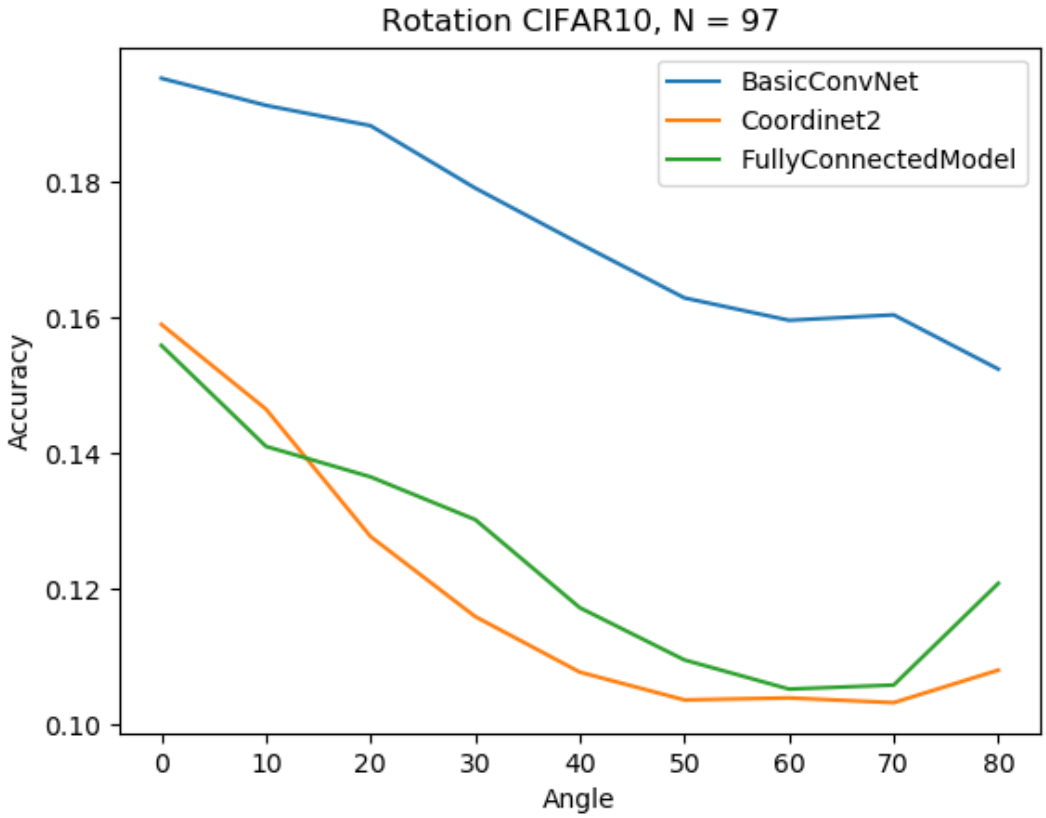



Rotation CIFAR10, N = 1562
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0
10
20
30
40
50
60
70
80
Angle


Rotation CIFAR10, N = 781
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0
10
20
30
40
50
60
70
80
Angle


Rotation CIFAR10, N = 390
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.22
0.20
0.18
0.16
0.14
0.12
0
10
20
30
40
50
60
70
80
Angle


Rotation CIFAR10, N = 195
BasicConvNet
Coordinet2
FullyConnectedModel
Accuracy
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0
10
20
30
40
50
60
70
80
Angle



Conclusions and Discussion

This work presents results that weakly suggest that the inductive bias of convolutional networks cannot be replicated by simply giving a model access to information about the relative location of pixels. This supports hypothesis 1, the standard interpretation of the inner workings of convolutional models. The generative model of Anokhin et al. (2020) may be best explained as using the attention-like ModFC layers to learn how to approximate a similar hierarchical representation of images to convolutional networks, taking advantage of huge dataset sizes and a very flexible model to overcome the lack of convolutional inductive biases. Future work should repeat the experiments of this paper enough times to confirm their significance.

References

Anokhin, I., Demochkin, K., Khakhulin, T., Sterkin, G., Lempitsky, V., & Korzhenkov, D. (2020). Image Generators with Conditionally-Independent Pixel Synthesis. *ArXiv:2011.13775 [Cs]*. http://arxiv.org/abs/2011.13775

Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., & Yosinski, J. (2018). An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. *ArXiv:1807.03247 [Cs, Stat]*. http://arxiv.org/abs/1807.03247

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature Visualization. *Distill*. 10.23915/distill.00007

Allen, C., & Hospedales, T. (2019). Analogies explained: Towards understanding word embeddings. In *International Conference on Machine Learning* (pp. 223-231). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.