Identifying invariant elements of Neural Networks to image transformations

Georgios Satyridis
STUDENT NUMBER: 2046944

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Supervisor: Giacomo Spigler
Second Reader: Grzegorz Chrupała

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2020

Preface

First and foremost, I would like to thank my supervisor Dr. Giacomo Spigler, who was extremely supportive throughout the course of the semester. His input and advice was of the utmost importance, greatly helping me choosing the right topic for my research. Under his supervision, I challenged myself academically and was able to exercise my interest in image analysis.

Moreover, I would like to thank my family and those of my friends who showed their support by helping me whenever I needed support. Leaving your career behind and starting anew is never easy and is rarely acknowledged. The encouragement I have received was surprisingly immense. I have made a lot of changes in both my professional and personal life whilst trying to find my own path and equilibrium in life and thus I am happy to be gifted with such a supportive family and friends.

Friends, especially those who support you in difficult situations, make your life richer. I would like to express my utmost gratitude to my friend and accomplished researcher Giovanni Rigazzi, who helped me with my relocation from Berlin to Tilburg last August. I will always remember the people who stood by my side and helped me go through any challenges, including those I have had to overcome this past year.

Identifying invariant elements of Neural Networks to image transformations

Georgios Satyridis

In this work, we attempt to identify some of the invariant elements of Neural Networks to geometric transformations using image data. Multilayer perceptrons are used as the benchmark for our sunsequent analysis. We observe that convolutional neural networks with pooling layers exhibit a degree of invariance to small magnitudes of translation. Data augmentation allows MLPs to generalize well, their performance when trained with rotated images is subpar though. Classifiers with global average and max pooling operations show a degree of invariance to shearing mapping.

1. Introduction

The objective of this work is not to address an existing practical problem but rather to identify and address knowledge gaps and also to contribute to the theoretical understanding of artificial neural networks, thus being an empirical work. To validate our hypotheses, we will conduct simulations on small and easily accessible datasets in order to make our experiments as reproducible as possible. The topic we will address in this research is the following; We aim to identify some of the elements of neural networks that exhibit some degree of invariance to geometric transformations by building on the work of previously published papers.

For the purposes of this work, we will focus on Neural Networks, but mainly on a specific category of Deep Neural Networks (DNN); Convolutional Neural Networks (CNN). Their importance has been evident and supported over the past years due to advancements in technology, as a result of powerful GPUs and cloud-based solutions (e.g. Colab, AWS), as well as to new techniques, such as Dropout (Srivastava et al. 2014). They are currently being used in various domains and have achieved state of the art results in natural language processing (Collobert and J. 2008), speech recognition (Mikolov et al. 2011) and medical image analysis (Shen, Wu, and Suk 2017) among others. In this work we will focus on their application in image classification tasks.

Image classification is a highly discussed topic in computer vision and has multiple practical applications. The success of CNNs in multiple vision tasks (Shen et al. 2019), has already been overstated by recent publications. Nevertheless, the theoretical understanding that we currently possess about DNNs is still rather limited (Lenc and Vedaldi 2014) (Poggio, Banburski, and Liao 2019), for instance it is still unknown to us how they extract features from input images (Iso, Shiba, and Yokoo 2018). This limited understanding has been the primary motivation of this work along with the drive to expand upon the findings of previously published papers.

The use of CNNs in image classification problems can be explained by their ability to learn high-dimensional and non-linear mappings by being trained with gradient descent, which make them ideal for such tasks (LeCun et al. 1998). ImageNet, which as at the time of writing contained over 14 million images, has been a significant contribution to the development of image classification research, due to the large number of training data and as a direct consequence has supported the development of DNNs. A significant breakthrough was made in 2012 by Krizhevsky along with his collaborators. They trained a deep convolutional neural network, by utilizing the computational power of GPUs, to classify 1.2 million images from ImageNet and they proved that such a deep architecture is capable of achieving state of the art results. (Krizhevsky, Sutskever, and Hinton 2012).

As it was previously mentioned, in this work our objective is to identify some of the elements of Neural Networks that have some degree of invariance to geometric transformations. The transformations that we will be applying to the data are translation, rotation, flipping and shearing. One notable observation in regards to CNNs is that they appear to have a built-in degree of translation-invariance, commonly on small magnitudes of translation, when pooling layers have been used as part of their architecture. A transition from average to max pooling has been observed in empirical studies, as a direct result of the increased performance of models using a max pooling operation (Boureau, Ponce, and LeCun 2010). CNNs have to rely on data augmentation, the process of artificially creating new data from the train set, and/or additional data, at least in principle, to learn other image transformations with success. (Krizhevsky, Sutskever, and Hinton 2012) (Han et al. 2020). With this method, they can approximate and even achieve human-level invariant recognition performance for familiar objects (Karimi-Rouzbahani, Bagheri, and Ebrahimpour 2017).

At this point, as invariance plays a central role in this work, we deem important to attempt to provide a short, albeit necessary definition of what invariance to image transformations in this specific domain is; Invariance is defined as the lack of significant effects that an image transformation has on the neural network's ability to learn from the image. Hence, the classifier's prediction does not change when a geometric transformation is applied to a given input. (Kamath, Deshpande, and Subrahmanyam 2020).

This work is organized in the following sections; We start by conducting a literature review, referring to recently published works which relate to our research. We then move on to the experimental setup, providing an overview of the datasets used in the experiments and explaining briefly the algorithms and techniques used to reach our results. In the following section we present the most intriguing results of our simulations. A discussion of the most important results and observations is then followed and lastly, we conclude by discussing about our methodologies and by presenting our findings, along with the key takeaways from this research.

2. Related Work

There has been extensive research which relates to the topic of this work and specifically to the effects of geometric transformations to the performance of neural networks in various domains, especially over the past decade. Such image transformations include but are not limited to scaling, rotation, flipping and translation. In this section of our work, we will present some interesting findings related to our research topic.

For image classification tasks, CNNs are the most common choice due to their robust performance, achieving state-of-the-art results. Moreover, it is worth noting that they do not require any feature engineering since they learn from the data that is fed into the network. This eliminates the need of manual feature engineering, requiring only limited preprocessing, shifting this task to computers (LeCun, Bengio, and Hinton (2015). Data augmentation is widely used to train networks to improve their generalization capability. Nonetheless, an insightful observation by Chen et al. (2019) states that models trained with the aforementioned method are not able to generalize well on a different set of augmented images.

To begin with, Han et al. (2020) investigated the degree of translation and scale-invariance in CNNs and Eccentricity-dependent Neural Networks (ENNs), with the latter being a modified version of CNNs with built-in scale-invariance and dependence of receptive field size on eccentricity. They trained both of them using data augmentation, which naturally is the only way to obtain "true" invariant representations in CNNs (Kauderer-Abrams 2017), by presenting them objects of different scale and position. As the authors expected, ENNs displayed higher degree of scale-invariance, suggesting the importance of a model's architecture. Additionally, they observed that CNNs develop example-based invariance, being limited to a specific set of data and thus cannot be extended to other sets.

An interesting study, from which we drew inspiration from, examined the effects of translation and rotation of images to the performance of artificial neural networks (Engstrom et al. 2017). They observed that even small rotations and translations greatly affected the robustness of the model. The lack of invariance to rotations of the input image is a topic that has been investigated in previous papers as well (Olivier et al. 2016). Translation invariance in CNNs is indeed present, albeit to a small degree. Only by using data augmentation can the network truly achieve high classification accuracy (Furukawa 2017).

The importance of data augmentation in supporting classifiers to achieve invariance in geometric transformations has been a highly discussed topic. To compute invariances in Neural Networks, Fawzi and Frossard (2015) proposed the Manitest method, based on the Fast Marching algorithm. With their method, they quantified invariance in CNN, which increased as the network's depth increased. Their results suggest that data augmentation is integral for classifiers to learn invariance from data. However, they did highlight that attaining invariance is task-specific and can be difficult in some tasks.

. In another study, Postma and van Noord N. (2016) extended on the work of Gluckman (2006) and proposed a multi-scale CNN method, which learns both scale-variant and scale-invariant representations from high resolution images. Additionally, they argued that scale-invariant and scale-variant representations in CNNs are beneficial to the performance of image recognition for tasks which include images of different scales and resolutions.

Lenc and Vedaldi (2014) suggested that CNNs are learned to be invariant towards some image transformations, such as re-scaling or rotation, however a standard CNN tends to not develop such invariant properties (Ngiam et al. 2010). Based on the findings of Shen et al. (2019), adding additional filters helps to alleviate this issue though. In the same work, they introduced SiCNN, a generalized version of CNN, comprised of multiple columns with each of them containing many convolutional layers with filter of various sizes. They used data augmentation to increase the network's robustness to scaling and flipping transformations. Their results indicated that their model yielded superior results compared to a typical CNN model.

Ngiam et al. (2010) introduced a Tiled CNN architecture, as an extension to a standard CNN model. In contrast to standard CNNs, a Tiled CNN does not tie all of the weights of the network together but rather leaves them untied on a local level, which they called "tiling". Based on their results, Tiled CNNs outperformed standard convolutional methods by allowing the network to learn invariances from unlabeled data. They found that a Tiled CNN has units which exhibit invariance to scaling and rotation. On the other hand, a standard CNN is highly unlikely to develop invariance to these specific geometric transformations.

The importance of pooling layers was examined by Scherer, Müller, and Behnke (2010), who addressed the effect of using pooling operations in capturing image invariances, using the Caltech-101 and the NORB datasets. Through their experiments they proved that for data derived from images, a max pooling operation yields significantly better results in capturing such invariances, compared to a subsampling pooling operation. This can be explained by the way a pooling layer operates. Pooling reduces the dimensionality of a feature map by calculating an aggregation function of small input regions, focusing on smaller areas of the image and thus compact the visual information (Boureau, Ponce, and LeCun 2010). They validated their model's performance on two additional datasets, which based on their results, was consistent to their previous findings.

Furthermore, Anselmi et al. (2014) introduced another approach that can help networks to learn invariant representations. A single labeled example, in this case a single signature that is invariant to transformations, can lead to significant results in feature learning. By extending a standard CNN architecture, Wang et al. (2017) created a multi-scale rotation-invariant CNN (MRCNN) model, which proved to be superior to conventional CNN architectures. They also suggested that deeper CNN models can extract high-level features in a better way than shallower models. Bengio and Lecun (2007) observed that networks with deeper architectures achieve better performance compared to those with shallower ones, by addressing the inefficiency of shallow architectures to accurately represent functions.

Moving away from CNNs, Goodfellow, Bengio, and Courville (2016) attempted to measure the invariances of stacked encoders using visual data, including images and video sequences. They also noted that by incorporating sparsity, networks tend to become more invariant to geometric transformations. They also observed that stacked autoencoders do not significantly improve the robustness of the network as its depth increases.

3. Experimental Setup

3.1 Data

In this research, we will focus on 3 different datasets, specifically the MNIST (LeCun 1998), which has been one the most widely-used datasets, the Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) and the CIFAR-10 (Krizhevsky 2009) datasets. All of the aforementioned datasets come preinstalled with the Keras (Chollet et al. 2015) open-source library. It is worth mentioning the classes of all 3 datasets are mutually exclusive. They are publicly available and have been chosen due to their size and ease of access, allowing for our results to be reproducible.

The MNIST dataset contains images of handwritten digits which range between the values 0 to 9, hence a total of 10 classes. In addition to, it comprises of 70,000 28x28 low resolution images in total. The dataset has already been split into a training and test set, containing 60,000 and 10,000 samples respectively. It is worth noting that all images are grayscale.

Fashion-MNIST was created to provide a better benchmark dataset compared to the original MNIST, having being described by researchers as too easy [1], and to eventually become its drop-in replacement. It contains 28x28 grayscale images of 70,000 fashion products, such as trousers and shirts, from 10 different categories. The training set of this dataset contains 60,000 images whereas the test set contains 10,000 images.

CIFAR-10 is a labelled subset of a dataset containing 80 million small images, which were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. It contains low resolution images, specifically 60,000 32x32 images, hence in this case the resolution of the images is slightly higher compared to the one of the two previously selected datasets. There are 10 different classes of various objects, which include different animals and machines used in transportation, while each class contains exactly 6,000 images. The training set is comprised of 50,000 samples, with the remaining 10,000 being in the test set. The most significant differentiating factor over the other two datasets, apart from the total number of data, is that the images are not grayscale but rather colour ones (RGB).

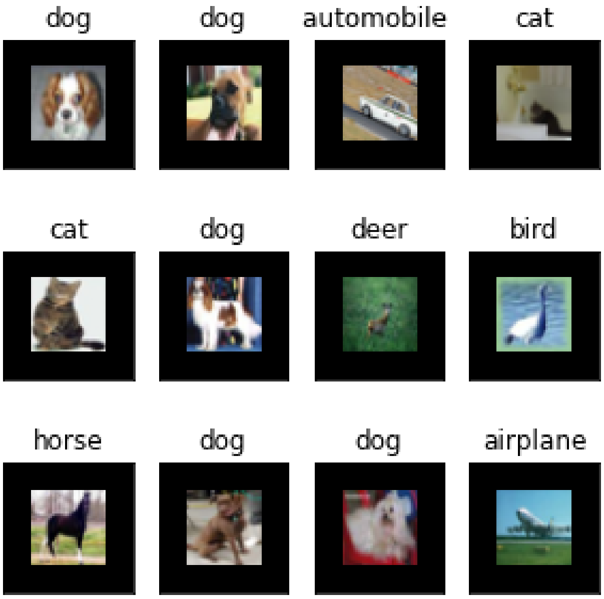
Table 1: Classes in the datasets along with their respective labels, apart from MNIST since its classes are the same as its labels (0 to 9).

Classes		
Label	Fashion-Mnist	CIFAR-10
0	T-shirt/top	Airplane
1	Trouser	Automobile
2	Pullover	Bird
3	Dress	Cat
4	Coat	Deer
5	Sandal	Dog
6	Shirt	Frog
7	Sneaker	Horse
8	Bag	Ship
9	Ankle Boot	Truck

[1] https://twitter.com/goodfellow_ian/status/852591106655043584

There was limited preprocessing involved for these datasets. We normalized the data by converting them to values within a range between 0 to 1 in order to allow the model to learn representations in a faster manner. The classes of each dataset were converted into binary matrices (0 and 1). We additionally extended the background size of all of the images. We achieved this by filling the empty space with zeros. In the MNIST and Fashion-MNIST datasets we doubled the size of the images to 56x56. We also increased the image sizes in the CIFAR dataset by approximately 80% to the same 56x56 size. The reason why we followed this particular approach was because we wanted to allow for different viewpoint observations (e.g. translation) without the objects, or any part of them, being moved outside of the boundaries of the image, in other words to provide sufficient space for allowing different geometric transformations to the images. This technique has been described by other studies (Engstrom et al. 2017) as the "black canvas" setting.

Figure 1: Randomly selected zero-padded images from the CIFAR-10 dataset along with their respective labels.



3.2 Method / Models

To perform our simulations, we have used the Keras open-source library, a high-level API built on top of the Tensorflow library (Abadi et al. 2015), which is, as of the time of writing, the most widely adopted Deep Learning platform (Chollet 2017). Additionally, we have made use of Python's scientific libraries, such as numpy (van der Walt, C., and G. 2011), while we have used OpenCV (Bradski 2000) and scikit-image (van der Walt S. et al. 2014) to apply various transformations to the images. All of the plots have been made with the matplotlib library (Hunter 2007). It is worth noting that the simulations of this work were completed using Google's Colaboratory harnessing a GPU runtime to hasten their time of completion.

To our best knowledge, we are not aware of any other published papers in which the exact same experiments have been conducted. To test our hypotheses, we have created 7 different models of varying depth for each dataset. Different architectures have been deployed, bringing the total number of models to 21. We have initially used a multilayer perceptron (MLP), which is going to be used as a benchmark for our subsequent simulations. The evaluation metric that is used is classification accuracy. To further test the MLP's performance, we employed data augmentation, while keeping the same architecture of the MLP model in each dataset, by presenting augmented data to the network. The transformations and the respective magnitude applied to the images used in data augmentation, cover the same range as the one that we have applied to the images in the test set. We expect that MLPs trained with data augmentation will achieve significantly better performance than the "standard" MLPs.

We have then defined 6 CNN models with different architectures. We do expect that the performance of MLP networks will be outperformed by the performance of CNNs. All of these neural networks are being trained with backpropagation, an iterative algorithm computing the gradient of the loss function with respect to each weight parameter.

In regards to the models' architecture, we have drawn inspiration from two different sources. Regarding MNIST, the network that our model has been loosely based on can be found on Tensorflow's tutorial (Abadi et al. 2015), which we have modified to fit the purposes of this research. The models used in the simulations for Fashion-MNIST and CIFAR-10 have been based on the architectures found on Keras' examples (Chollet et al. 2015), and changes have been made to each model respectively.

At this part we will present the networks that we have used for the simulations in more detail. Apart from the standard MLP model, we have defined a total of 6 CNN models; a standard CNN, a CNN with max pooling and another model with average pooling layers. We also used a CNN with dropout layers of medium rate (0.3) and two additional models with a global max pooling and a global average pooling operation respectively. All of the networks include fully connected layers at the end of the architecture, with either 128 or 256 (MLPs in Fashion-MNIST & CIFAR-10) units. In contrast to the MLPs, the CNN models include 2 fully connected layers, thus these networks have densely connected architectures.

There are a few commonalities between the models deployed that are worth to be addressed at this point. For CNNs in the CIFAR-10 and Fashion-MNIST datasets, we have stacked the convolutional layers and have maintained the same number of filters for 2 layers in a row and subsequently increasing them by 2 (e.g. 32, 64). In all CNNs we have used a three by three kernel and kept the default step size of 1 in each convolutional layer. Additionally we have used padded convolutions, which add pixels, and specifically pad zeros, to the edges of the image to ensure that the pixels of the output feature map have the same shape as the ones of the input. Padding allows us to maintain the same spatial dimensions between the input and the output layers. All max and average pooling layers use a 3x3 kernel with a stride parameter of 1.

There are 10 classes in each dataset that we are trying to predict, hence the output layer of all of the models contains exactly 10 units. The networks have been trained using the same optimization algorithm, Adam at its default learning rate, as well as the cross-entropy loss function. Furthermore, we have split the data between train, validation and test sets. We have then iterated the models over 10 epochs and chosen the best weights.

To introduce non-linearity to the networks, the rectified linear units (ReLU) activation function has been chosen and used after all convolutional and dense layers, since it solves the vanishing gradient problem, apart from the output layer where we have used softmax, which is the most appropriate activation function to use for multi-class classification problems (Bridle, F.F., and Hérault 1989).

The learning algorithms are being trained using a set of images to which no transformations have been applied to, while leaving a development set aside to validate each model's performance. Consequently, they are being evaluated on various test sets, to which image transformations have been applied to. Since we are dealing with relatively small datasets, we expect to observe a degree of overfitting, especially in the models that no significant regularization techniques have been applied to. Regardless, we are using the best weights of each model, where the losses converge.

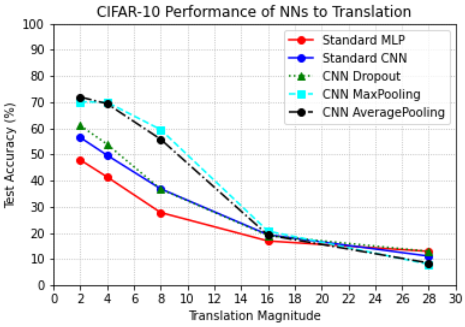
The transformations applied to the images include flipping (horizontal and vertical) as well as different magnitudes of rotation, translation and shearing. The motivation of exploring the effects of shearing mapping, an affine transformation, to neural networks stems from the limited research conducted. It needs to be noted that regarding translation, each image has been swifted along the x and y axis by the same level of magnitude. One of our objectives is to address whether neural networks have the ability to identify objects as we move them closer to the edges of the image. This approach can be reflected in some of the transformations applied to the test set.

4. Results

In this section, we will briefly present the most interesting findings of the simulations that we have conducted, including the results of the MLP models after training them on augmented images. Only the CIFAR-10 plots will be shown here, while the rest of the plots can be found in the appendix.

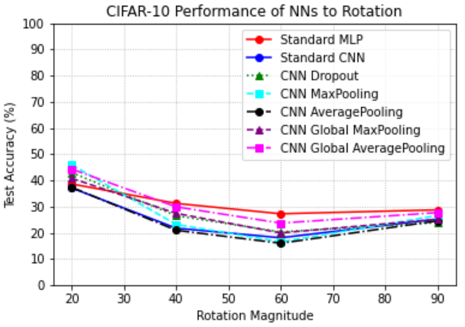
The plots showing the networks' performance on translation indicate that CNNs with pooling layers have a higher degree of translation-invariance, especially after small pixel swifts, in contrast to the rest of the models. Both global max and average pooling models have been excluded from all plots indicating the performance on translation, since it does not change across all positional swifts. Models comprising of a global pooling layer are generally more robust to spatial translations of the input (Lin, Chen, and Yan 2013), which explains our results.

Figure 2: CIFAR-10 - Performance of NNs to translation.



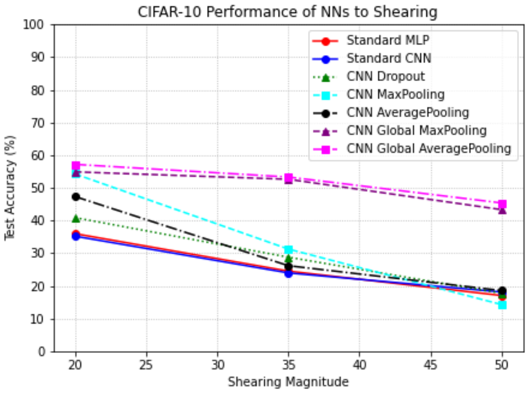
The performance of the networks on various degrees of rotation follows a downward trend. A drop in performance of around 50% is observed in the images of the CIFAR-10 dataset, whereas the decrease in the remaining datasets is significantly steeper. The performance of MLPs is surprising and has to be addressed though. Their test accuracy is not greatly affected by the magnitude of rotation, varying between 30-40%. In the remaining datasets however, their robustness is diminished.

Figure 3: CIFAR-10 - Performance of NNs to rotation.

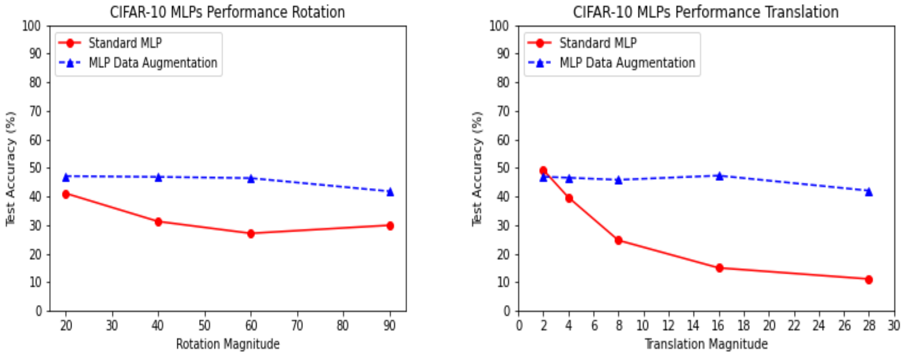


The effect of shearing mapping to neural networks has been investigated as well. We do observe a small degree of invariance in the models with a global pooling layer. These findings are consistent across all datasets employed. Generally, the rest of the models do not perform well when introduced to shearing transformations, with a clear deterioration in their accuracy.

Figure 4: CIFAR-10 - Performance of NNs to shearing.



The importance of data augmentation is evident in our results. Models trained on augmented images, separately trained on a given set of transformed images, show a higher degree of translation and rotation invariance than the models trained on non-augmented data, hence they outperform standard MLP models. The following plots depicts a clear deterioration in performance of multilayer perceptrons that have been trained on non-augmented data.



(a) Rotation

(b) Translation

Figure 5: MLPs Performance to Rotation and Translation with standard architecture and Data Augmentation

The results to horizontal and vertical flipping are not presented in this section but can be rather found at part 6 of the appendix. Most models show some degree of invariance to vertical flipping, which is more evident in the models with an average and max pooling operation. They do not exhibit the same properties when presented with horizontally flipped images though, indicating that even state-of-the-art classifiers do not possess such transformation invariant elements.

5. Discussion

We expected that convolutional layers would introduce elements to the networks that would enable them to be to a certain extent translation invariant. Consistent with empirical findings, models with pooling layers boost the invariant effects to small magnitudes of translation. The corresponding plots indicate that CNNs with pooling layers have a higher degree of translation-invariance in contrast to the other models. We do observe, however, a deterioration in their performance after shifting the objects by at least 4 pixels.

The networks perform poorly when evaluated on rotated images of various degrees. This is not the case with the MNIST dataset, however due to its simplicity compared to the other datasets, its results cannot be considered as conclusive. We are of the opinion that the degree of rotation-invariance in neural networks increases when transformed representations are encoded to the model. The MLP model shows promising results, being slightly more invariant to rotation in the CIFAR-10 dataset. We expect that further tests on other datasets, using an architecture different to ours with a varying number of layers and architectures, would yield more robust results since we followed a simple approach when evaluating the performance of MLPs.

An interesting finding comes from the simulations ran on transformed images using shearing mapping. Networks trained using a global pooling operation have significant shear-invariant properties. It is more evident for models encompassing a global average pooling layer since the performance of models containing a such operation are superior to the ones with a global max pooling layer. We expected vice-versa results and believe that further investigation needs to be made.

The performance of models trained on augmented data were not in line with our expectations. We did expect that data augmentation would increase the robustness of the MLPs to geometric transformations. Such networks trained with swifted images along both axes exhibit strong invariance to translation. On the other hand, the same networks trained on rotated images show a decay in their performance, thus being less rotation-invariant than we had initially expected. The drop in performance is more evident in the results obtained for the MNIST and Fashion-MNIST datasets.

A surprising result is the performance of CNNs when evaluating vertically flipped image data. The networks trained using pooling operations have stronger invariant elements than the rest. However, the results are not consistent across all datasets, which indicates the role of adopting domain-specific approach.

Our findings suggest that including Dropout layers, which drop random units during training, in the architecture of CNNs does not play a pivotal role which would allow them to learn invariant representations. In other words, Dropout neither supports nor enhances the ability of CNNs to learn invariant representations to geometric transformations.

6. Conclusion

In this work, we examined the robustness of classifiers to four different geometric transformations. Our results were on par with the literature when evaluating their robustness on translation invariance. Convolutional layers are capable of capturing invariance in neural networks which is further intensified when a pooling operation, particularly max pooling, is introduced to the architecture of the network. MLPs become translation-invariant only when trained with augmented images. On the other hand, MLPs that are trained with data augmentation appear to not learn rotation-invariant representations as effectively as CNNs, even though they do seem to capture some invariant elements.

The robustness of global pooling operations to geometric transformations should be further investigated, potentially with models which are not comprised of fully connected layers, since the motivation of having such operations was driven by the need of eliminating the use of the aforementioned layers. We would further encourage research to be undertaken to investigate the reason that drives models which use a global average pooling operation to be more invariant to rotation and shearing compared to a global max pooling operation.

A particular highlight is the invariance shown by networks which contain max and average pooling layers to vertical flips. It would be interesting to conduct further simulations on additional image datasets, to observe whether this level of robustness remains.

Finally, results can be inconsistent across different domains and approaching an object recognition task should be domain-specific and tailored to the data at hand to address any potential peculiarities. We are motivating the idea to run additional simulations using a varying number of filters, kernels and strides. This would allow us to evaluate the robustness of state-of-the-art models to spatial transformations, using bigger and more complex datasets.

References

Abadi, M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S, Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., and Zheng X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. tensorflow.org.

Anselmi, F., Leibo J. Z., Rosasco L., Mutch J., Tacchetti A., and Poggio T. 2014. Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning? Technical report, MIT Center for Brains.

Bengio, Y. and Y. Lecun. 2007. Scaling learning algorithms towards AI. Large-Scale Kernel Machines.

Boureau, Y.L., J. Ponce, and Y. LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. pages 111–118.

Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Bridle, John S., Soulié F.F., and J. Hérault. 1989. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition neurocomputing: Algorithms, architectures and applications. In *NATO ASI Series (Series F: Computer and Systems Sciences)*, page 227–236, Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-76153-9_28.

Chen, W., L. Tian, L. Fan, and Y. Wang. 2019. Augmentation invariant training. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

Chollet, F. 2017. *Deep Learning with Python*. Manning.

Chollet, F. et al. 2015. Keras. `https://keras.io`.

Collobert, R. and Weston J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*.

Engstrom, L., Tran B., Tsipras D., Schmidt L., and Madry A. 2017. Exploring the landscape of spatial robustness. arXiv:1712.02779.

Fawzi, A. and P. Frossard. 2015. Manitest: Are classifiers really invariant? arXiv:1507.06535.

Furukawa, H. 2017. Deep learning for target classification from sar imagery: Data augmentation and translation invariance. arXiv:1708.07920.

Gluckman, J. 2006. Scale variant image pyramids. volume 1, pages 1069 – 1075. 10.1109/CVPR.2006.265.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Han, Y., G. Roi, G. Geiger, and Poggio T. 2020. Scale and translation-invariance for novel objects in human vision. *Sci Rep 10*. https://rdcu.be/b33ln.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95. 10.1109/MCSE.2007.55.

Iso, S., S. Shiba, and S. Yokoo. 2018. Scale-invariant feature extraction of neural network and renormalization group flow. *Physical Review E*, 97(5). arXiv:1801.07172.

Kamath, S., A. Deshpande, and K. M. Subrahmanyam. 2020. Invariance vs. robustness trade-off in neural networks. arXiv:2002.11318.

Karimi-Rouzbahani, H., N. Bagheri, and R. Ebrahimpour. 2017. Invariant object recognition is a personalized selection of invariant features in humans, not simply explained by hierarchical feed-forward vision models. *Scientific Reports*, 7.

Kauderer-Abrams, E. 2017. Quantifying translation-invariance in convolutional neural networks. arXiv:1801.01450.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

LeCun, Y. 1998. The mnist database of handwritten digits. yann.lecun.com/exdb/mnist/.

LeCun, Y., Y. Bengio, and G. Hinton. (2015). Deep learning. *Nature*, 521: 436–44. 10.1038/nature14539.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324.

Lenc, K. and A. Vedaldi. 2014. Understanding image representations by measuring their equivariance and equivalence. arXiv:1411.5908v2.

Lin, M., Q. Chen, and S. Yan. 2013. Network in network. arXiv:1312.4400v3.

Mikolov, T., Deoras A., Povey D., Burget L., and Cernocky J. 2011. Strategies for training large scale neural network language models. *2011 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2011, Proceedings*.

Ngiam, J., Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A.Y. Ng. 2010. Tiled convolutional neural networks. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., pages 1279–1287. paper.

Olivier, M., Veillard A., Lin J., Petta J., Chandrasekhar V., and Poggio T. 2016. Group invariant deep representations for image instance retrieval. arXiv:1601.02093.

Poggio, T., A. Banburski, and Q. Liao. 2019. Theoretical issues in deep networks: Approximation, optimization and generalization. arXiv:1908.09375.

Postma, E. and van Noord N. 2016. Learning scale-variant and scale-invariant features for deep image classification. arXiv:1602.01255v2.

Scherer, D., A. Müller, and S. Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *Artificial Neural Networks – ICANN 2010*, pages 92–101, Springer Berlin Heidelberg, Berlin, Heidelberg. 10.1007/978-3-642-15825-4_10.

Shen, D., G. Wu, and H. I. Suk. 2017. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1): 221–248. PMID: 28301734.

Shen, Xu, Xinmei Tian, Shaoyan Sun, and Dacheng Tao. 2019. Patch reordering: a novel way to achieve rotation and translation invariance in convolutional neural networks. arXiv:1911.12682.

Srivastava, N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1): 1929–1958.

van der Walt, S., Colbert S. C., and Varoquaux G. 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2): 22–30.

van der Walt S., Schönberger J. L., Nunez-Iglesias J, Boulogne F., Warner J. D., Gouillart E. Yager N., and Yu T. 2014. scikit-image: image processing in python. *PeerJ 2:e453*. `https://doi.org/10.7717/peerj.453`.

Wang, Q., Y. Zheng, G. Yang, W. Jin, X. Chen, and Y. Yin. 2017. Multi-scale rotation-invariant convolutional neural networks for lung texture classification. *IEEE Journal of Biomedical and Health Informatics*, PP:1–1.

Xiao, H., K. Rasul, and R. Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*. arXiv:1708.07747v2.

Appendix A: Rotation Plots

Figure 1: MNIST - Performance of the models to rotation.

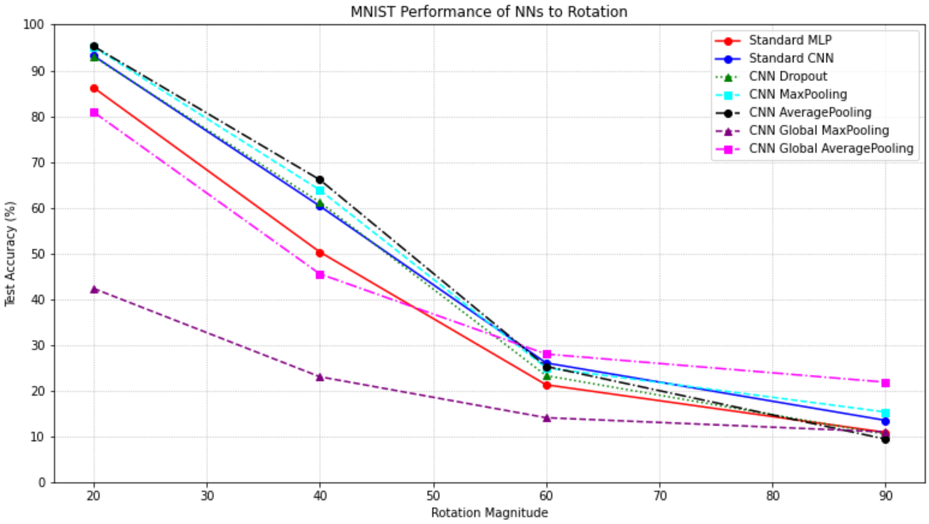
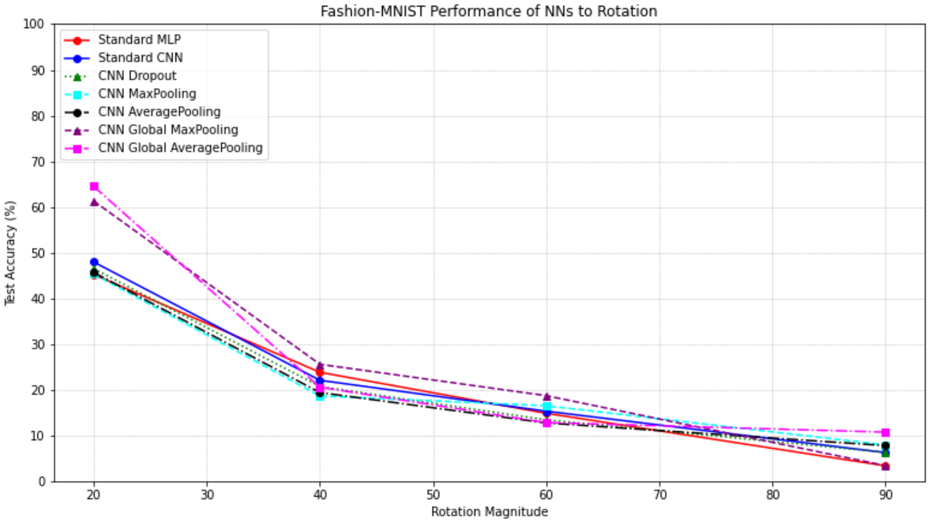


Figure 2: Fashion-MNIST - Performance of the models to rotation.



Appendix B: Translation Plots

Figure 1: MNIST - Performance of the models to translation.

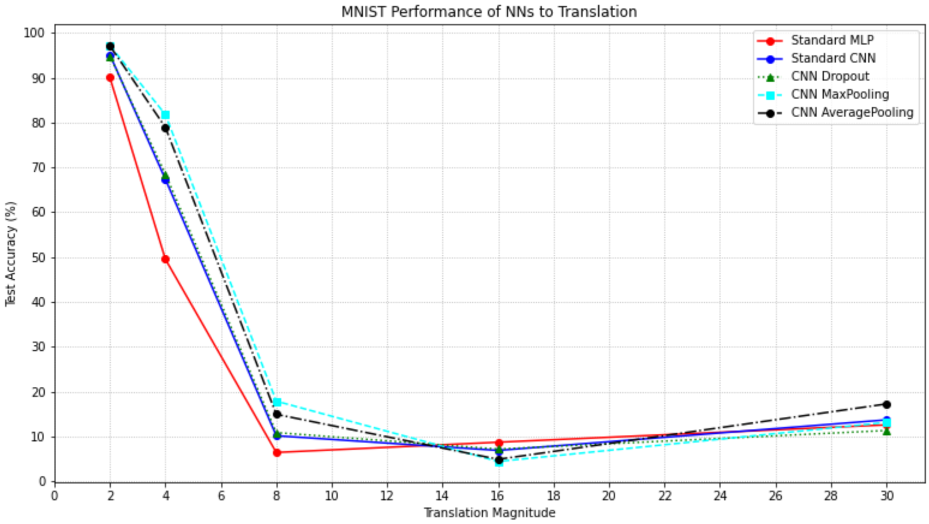
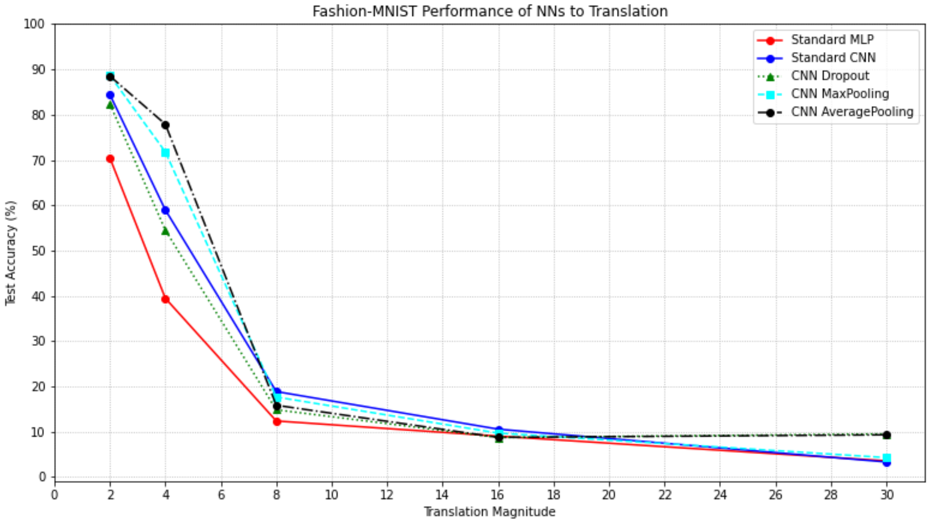


Figure 2: Fashion-MNIST - Performance of the models to translation.



Appendix C: Standard MLP and MLP with Data Augmentation Plots

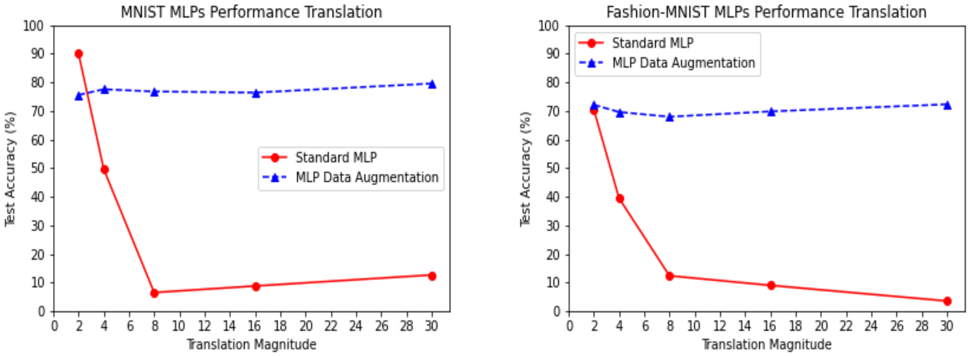


Figure 1: MNIST

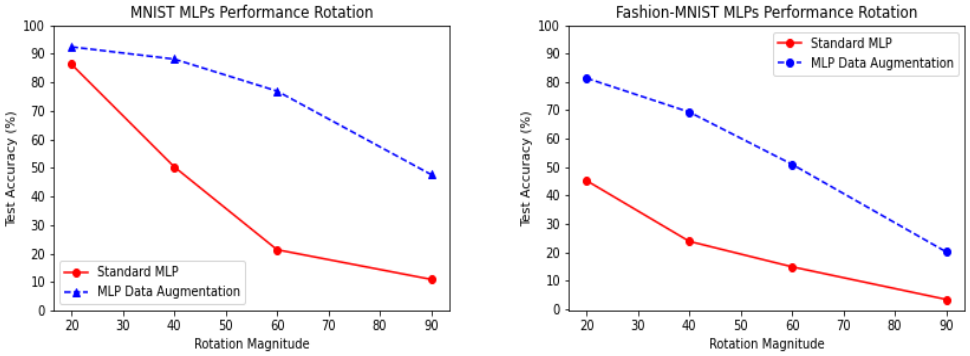


Figure 2: Fashion-MNIST

Appendix D: Shear Plots

Figure 1: MNIST - Performance of the models to shearing.

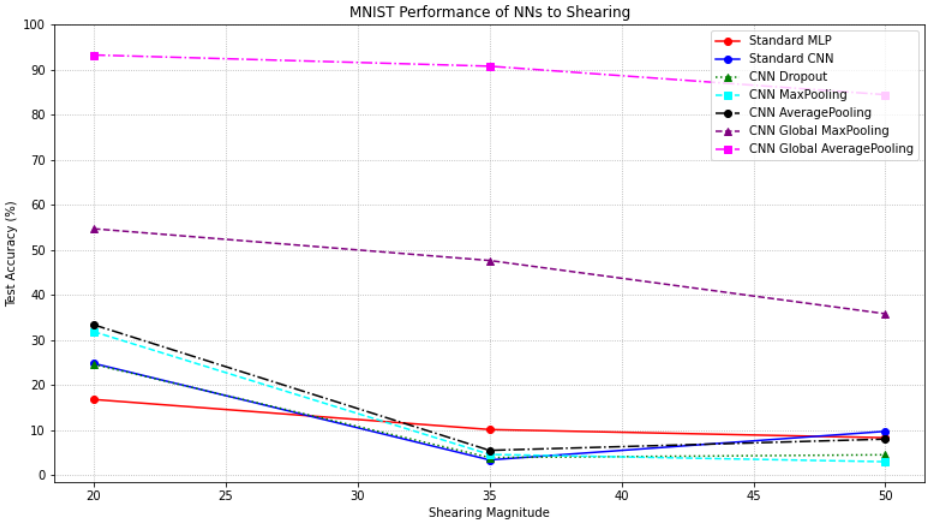
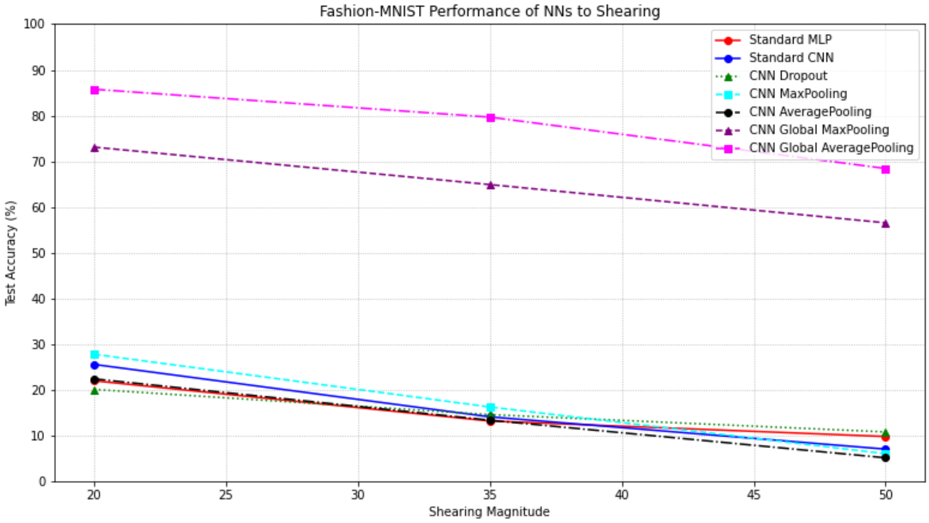
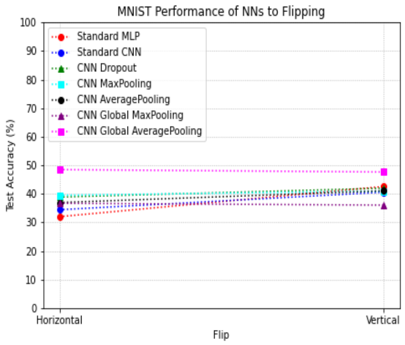
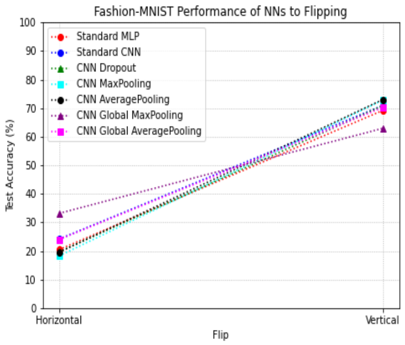


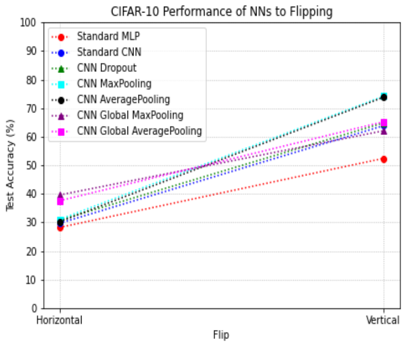
Figure 2: Fashion-MNIST - Performance of the models to shearing.



Appendix E: Horizontal & Vertical Flip Plots


MNIST Performance of NNs to Flipping
Test Accuracy (%)
Horizontal
Vertical
Flip
Standard MLP
Standard CNN
CNN Dropout
CNN MaxPooling
CNN AveragePooling
CNN Global MaxPooling
CNN Global AveragePooling


Fashion-MNIST Performance of NNs to Flipping
Test Accuracy (%)
Horizontal
Vertical
Flip
Standard MLP
Standard CNN
CNN Dropout
CNN MaxPooling
CNN AveragePooling
CNN Global MaxPooling
CNN Global AveragePooling


CIFAR-10 Performance of NNs to Flipping
Test Accuracy (%)
Horizontal
Vertical
Flip
Standard MLP
Standard CNN
CNN Dropout
CNN MaxPooling
CNN AveragePooling
CNN Global MaxPooling
CNN Global AveragePooling