UPPSALA
UNIVERSITET

Rotation equivariant and invariant CNNs

- implementation and performance analysis

Peipei Han, Rebecca Stenberg

Project in Computational Science: Report

January 2020

PROJECT REPORT

Institutionen för informationsteknologi

Abstract

Standard Convolutional Neural Networks (CNNs) are naturally translation invariant since correlation or convolution operations are shift invariant. However, they are not rotation equivariant nor invariant. In this report, three rotation equivariant and/or invariant methods are evaluated to testify that they do improve image classification and can be applied to help detecting oral cancer.

Contents

1 Introduction

The successful application of Convolution Neural Networks (CNNs) in image classification tasks can be attributed largely to the fact that convolution or correlation operations are translation-invariant. However, convolution or correlation operations do not exhibit invariance to rotations. If such invariance is not explicitly encoded, the network must learn it from the data, usually by augmenting the training input images with rotations, which requires more parameters and thereby increases susceptibility to overfitting.

In this report, we introduce three CNNs into which rotation invariance can be explicitly encoded: Group Equivariant Convolutional Networks (G-CNNs) (Cohen and Welling, 2016), Conic Convolution and Discrete Fourier Transform Network (CFNet) (Chidester et al., 2019) and Harmonic Networks (H-nets) (Worrall et al., 2016). G-CNNs maintain the property of rotation equivariance throughout the convolution layers of the network and pools for each filter to encode invariance. CFNet proposes two innovations for CNNs: conic convolution layers and 2D-discrete-Fourier transform (2D-DFT). H-Nets replace the regular CNN filters with circular harmonics. Improvement in classification accuracy of G-CNNs, CFNet and H-Nets over a standard CNN, is shown in the classification of rotated images of MNIST.

We further apply these methods to detect oral cancer. As we all known, the prognosis for patients diagnosed with oral cancer is worse than for the average cancer disease and one of the most effective ways to decrease the mortality is by early detection (Öhman, 2019) . According to World Health Organization 657,000 new cases of oral cancer are estimated each year and more than 330,000 deaths. Screening cells from the oral cavity and classifying them as cancerous or healthy is one way to detect cancer at an early stage. CNNs with improved performance for rotated images could be a help when trying to increase the accuracy for diagnosing cancer.

The three models evaluated in this report are described in Section 2. Section 3 describes implementation details and results of our experiments on rotated MNIST and Oral Cancer Dataset. We discuss the three models based on the results in Section 4. Source code for the implementation is available on `https://github.com/Carolinehan/CS-Project`.

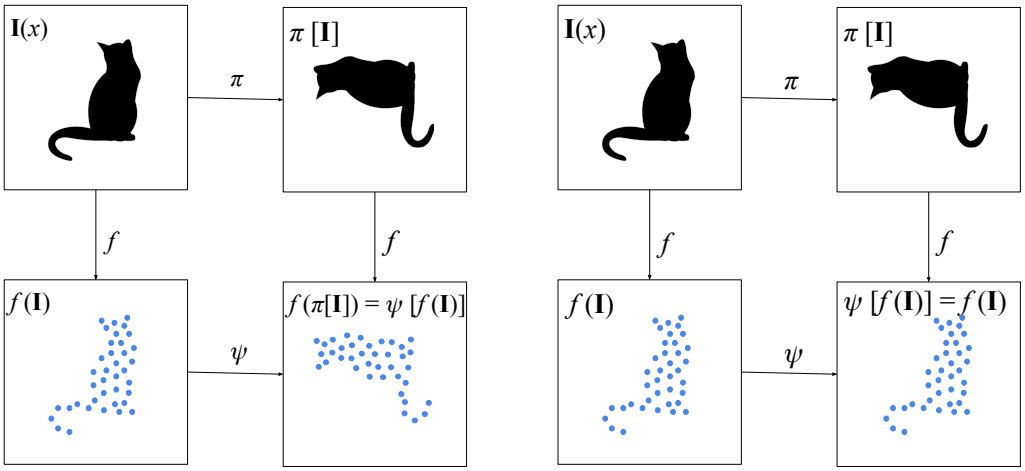
2 Background

In this section, we first explain equivariance and invariance, then describe the three rotation equivariant and/or invariant methods. G-CNNs, CFNet and H-Nets all encode rotation equivariance and invariance through filters instead of augmenting

training data. However, in G-CNNs, rotated filters are convolved across entire input feature maps, in CFNet, rotated filters are only convolved along radial, conic regions of the input feature map and H-Nets use angle frequency instead of discrete degrees. Theoretically, it is expected that H-Nets perform best while slowest and for CFNet to perform worse but faster than G-CNNs. All of them are expected to perform better and slower than standard CNNs.

2.1 Equivariance and invariance

A transformation is equivariant if its application on the input results in a predictable change of the (features of the) output. In Figure 1(a) an original image $\mathbf{I}(x)$ of a cat and a copy of the original image after a rotation by π rad are shown. Their corresponding feature maps $f(\mathbf{I})$ and $f(\pi[\mathbf{I}])$ are represented with blue circles and the difference can be described by the transformation ψ. Invariance is a special case of equivariance where the transformation ψ is the identity transformation. In this case the feature vector remains constant for invariant transformations as shown in Figure 1(b).



(a) (b)

Figure 1: (a) Rotation equivariance (b) Rotation invariance (Lohrelei)

2.2 Group Equivariant Convolutional Networks(G-CNNs)

The insight of G-CNNs is to apply convolution over transformation groups including rotation, translation and flips, thereby maintaining equivariance throughout the convolutional layers. G-CNNs represent feature maps in 2D space $\mathbb{Z}^2$ of integers, or pixel locations in the case of images as a function $f : \mathbb{Z}^2 \to \mathbb{R}^K$ and filters as $\phi : \mathbb{Z}^2 \to \mathbb{R}^k$ where K denotes dimension. The functions are operated on symmetry groups, thereby G-CNNs achieving equivariance for rotation transformations. A symmetry group has the following properties:

- The composition of two symmetry transformations g and h is another symmetry transformation.
- The inverse g^{-1} of any symmetry g is also a symmetry, and composing it with g gives the identity transformation e.

The expression for convolution of a filter over a feature map in a standard CNN is given by

$$f * \phi(x) = \sum_{k=0}^{K-1} \sum_{z \in \mathbb{Z}^2} f_k(z)\phi_k(z - x),$$

where the filter is translated over the image and the inner product is computed at each translation (Chidester et al., 2019). G-CNNs replace the shift by a more general transformation from some group G, such as the group G = p4 of 90° rotations, or G = p4m which additionally includes reflection. p4 group can be parameterized by g(r) acting on points in Z^2 by matrix multiplication for a given point $x = (u^{'}, v^{'}) \in \mathbb{Z}^2$ as

$$g(r, u, v) = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix},$$

where $0 \leq r < 4$ and $(u, v) \in Z^2$ (Cohen and Welling, 2016).
In the first layer of G-CNNs, by replacing shift with group G, we get the G-Convolution:

$$[f \star \phi](g) = \sum_{y \in Z^2} \sum_{k} f_k(y)\phi_k\left(g^{-1}y\right).$$

For all layers after the first, the convolution operation becomes

$$[f \star \phi](g) = \sum_{h \in G} \sum_{k} f_k(h)\phi_k\left(g^{-1}h\right).$$

In the final layer, a group-pooling layer is used to ensure that the output is either invariant (for classification tasks) or equivariant as a function on the plane (for segmentation tasks, where the output is supposed to transform together with the input) (Veeling et al., 2018).

2.3 Conic Convolution and Discrete Fourier Transform Network(CFNet)

G-CNNs consider both local and global rotation equivariance in convolutional networks. By delaying the imposition of invariance till the final, fully-connected layers, as much global structure as possible is thereby incorporated. However, by applying pooling on individual filters, potentially valuable mutual rotational information across filter responses is therefore lost. CFNet, by having all the advantages of G-CNNs, captures global rotation invariance better. It consists of two innovations: conic convolution layer and 2D-Discrete Fourier Transform as described in following. In Figure 2 we demonstrate the overall architecture of CFNet.

2.3.1 Conic Convolution Layer

Rather than convolving each filter across the entire image, rotated filters are convolved along radial, conic regions of the input feature map. This is accomplished by convolving the input with each filter, rotated by multiple of $\pi/2R$, for $R \in Z_{>0}$, over corresponding conic regions of the domain. As G-CNNs, CFNet also considers feature maps as functions over 2D space. The feature map is partitioned into conic regions $\{\mathcal{C}_r\}_{r=0}^{4R-1}$. The boundaries of conic regions are denoted by $\{\mathcal{B}_r\}_{r=0}^{4R-1}$. They can be expressed

$$\mathcal{C}_r = \left\{(x,y) \in \mathbb{Z}^2 : \theta_r < \operatorname{arccot}(x/y) + \pi\mathbb{I}(y<0) < \theta_{r+1}\right\},$$
$$\mathcal{B}_r = \left\{(x,y) \in \mathbb{Z}^2 : \operatorname{arccot}(x/y) + \pi\mathbb{I}(y<0) = \theta_r\right\},$$

where $\theta_r = \frac{2\pi r}{4R}$ and $\mathbb{I}(\cdot)$ is the indicator function (Chidester et al., 2019).

2.3.2 2D-Discrete Fourier Transform(2D-DFT)

Normally, after the final convolutional layer, fully connected layers will be applied to combine responses from all the filters. However, fully connected layers do not maintain rotation equivariance or invariance properties since the output of the last convolutional layer is usually a vector without spatial information. Rather than encoding invariance for each filter separately, as in G-CNNs, CFNet instead transforms the collective filter responses to a space in which rotation becomes circular shift so that the 2D-DFT can be applied to encode invariance. The primary advantage of the 2D-DFT as an invariant transform is that each output node is a function of every input node, and not just the nodes of a particular filter response, thereby capturing mutual information across responses. In this layer, feature maps are represented as tensors, rather than functions as DFT is defined only for finite-

length signals.

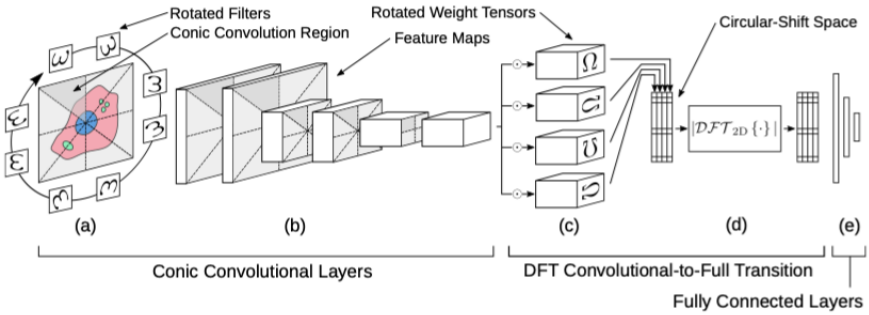


Figure 2: The overall architecture of the proposed rotation-invariant CNN. (a) Filtering the image by various filters $\{\omega\}$ at rotations in corresponding conic regions preserves rotation-equivariance. (b) Subsequent convolutional feature maps are filtered similarly. Rotation-invariance is encoded by the transition from convolutional to fully-connected layers, which consists of (c) element-wise multiplication and sum, denoted by $\odot$, with rotated weight tensors $\{\Omega\}$, transforming rotation to circular shift, and (d) application of the magnitude response of the 2D-DFT to encode invariance to such shifts. (e) This output is reshaped and passed through the final, fully-connected layers (Chidester et al., 2019).

2.4 Harmonic networks

Harmonic networks (H-nets) described in (Worrall et al., 2016) hard-bake rotation equivariance into the structure by replacing regular convolutional filters with circular harmonics. This way H-nets exhibit equivariance for 360°rotation and patch-wise translation. A filter from the circular harmonics family can be expressed

$$\mathbf{W}_m(r, \phi; R, \beta) = R(r)e^{i(m\phi+\beta)}, \tag{1}$$

where r, ϕ are the polar coordinates of the image, $R : \mathbb{R}_+ \to \mathbb{R}$ the radial profile, $\beta \in [0, 2\pi)$ a phase offset term and $m \in \mathbb{Z}$ the rotation order. The radial profile is a function controlling the filter's shape and the filter has orientation selectivity due to the phase offset term. The radial profile and the phase offset are the learnable parameters. The rotation order corresponds to an angular frequency and each circular harmonic picks up a different rotation order in the input making it possible to process each in a separate stream. To move between streams cross-correlation of the rotation order that equals the difference between the two streams is used. This means the streams can share information at each layer which allows

for rotation equivariant deep features more complex than the circular harmonic itself.

Since the circular harmonic filters are defined in the polar domain and the images in the dataset are sampled on a rectangular grid in 2D we anti-alias the input to a discretized layer which is done with a Gaussian blur. The filter after being resampled can be described with

$$W(\mathbf{x}_i) = \sum_j g_i(\mathbf{r}_j) W(\mathbf{r}_j), \tag{2}$$

where $\mathbf{x}_i$ is a pixel on the rectangular grid and $g(\mathbf{x}_j) \propto e^{-||\mathbf{r}_i - \mathbf{x}_j||_2^2/(2\sigma^2)}$. If $\mathbf{R}_j$ is a radial tensor, $\mathbf{I}$ the identity matrix and $[\mathrm{cos}m\mathbf{\Phi}_{r_j}, i\mathrm{sin}m\mathbf{\Phi}_{r_j}]^T$ an angular tensor then

$$W_m(\mathbf{r}_j) = \sum_j^I R(r_j) \begin{bmatrix} \mathbf{I}cos\beta & -\mathbf{I}sin\beta \\ \mathbf{I}sin\beta & \mathbf{I}cos\beta \end{bmatrix} \begin{bmatrix} cosm\mathbf{\Phi}_{r_j} \\ isinm\mathbf{\Phi}_{r_j} \end{bmatrix}. \tag{3}$$

3 Experiments

In this section, we describe rotated MNIST and Oral Cancer Dataset and how they are pre-processed for our experiments. We present the selected architecures and training parameters for the three rotation equivariant and/or invariant CNNs we discussed in the background section together with a standard CNN model. We first evaluate these models on rotated MNIST. Then we optimize some of the four models to evaluate on Oral Cancer Dataset.

Unless otherwise specified, our models are implemented in Tensorflow 1.14 and Python2 using GPU Nvidia GeForce GT 730.

3.1 Rotated MNIST

The rotated MNIST dataset (public_static_twiki) consists of 10000 training images, 2000 validation images and 50000 test images. The images are variations of the handwritten digits in the common MNIST dataset where the numbers have been rotated with uniformly generated angles between 0 and 2π, see Figure 3 for a sample. The images have the size 28x28x1 and each image belongs to one of the ten classes corresponding to the digit.

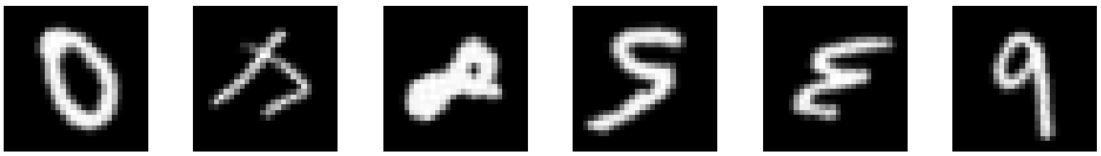


Figure 3: Samples from the rotated MNIST dataset.

We normalize the whole dataset by mean and standard deviation calculated over training set, so that the data is centered around 0 and has a unit standard deviation with an aim to improve convergence speed and accuracy.

To verify if rotation equivariant and/or invariant methods improve rotated MNIST classification, we reduce impacts from other aspects except the methods themselves, such as hyper-parameters and number of layers. We limit all four models to consist of nine layers and adopt the same hyper-parameters:

- Batch size: 100
- Epochs: 100
- Adaptive learning rate: Initialized with 0.001 and decreased if no improvement after 12 epochs in validation set.
- Optimizer: Adam

3.1.1 Standard CNN

The architecture consists of seven convolution layers, each with 10 3x3 filters, one max pooling layer, and one fully connected layer as shown in Figure 4(a). The max pooling layer uses a filter of size 2x2 with stride 2. We apply batch normalization and Relu activation function after every convolution layer.

3.1.2 G-CNN

As shown in Figure 4(b), G-CNN model has a similar architecture to the standard CNN model except it uses G-Convolution layer instead of convolution layer and the filter size of the last G-Convolution layer is 4x4 as opposed to 3x3.

3.1.3 CFNet

CFNet also has nine layers in total consisting of six conic convolution layers, one max pooling layer, and one fully connected layer. Different from standard CNN and G-CNN models, CFNet has a 2D-DFT transition layer before the fully connected layer as Figure 4(c) shows.

3.1.4 H-Nets

H-Nets contain seven cross-correlated layers and two mean-pooling as shown in Figure 4(d). They have the characteristic streams, in this setup there are two rotation orders, $m = 0$ and $m = 1$. The polar filters have the size 5x5.

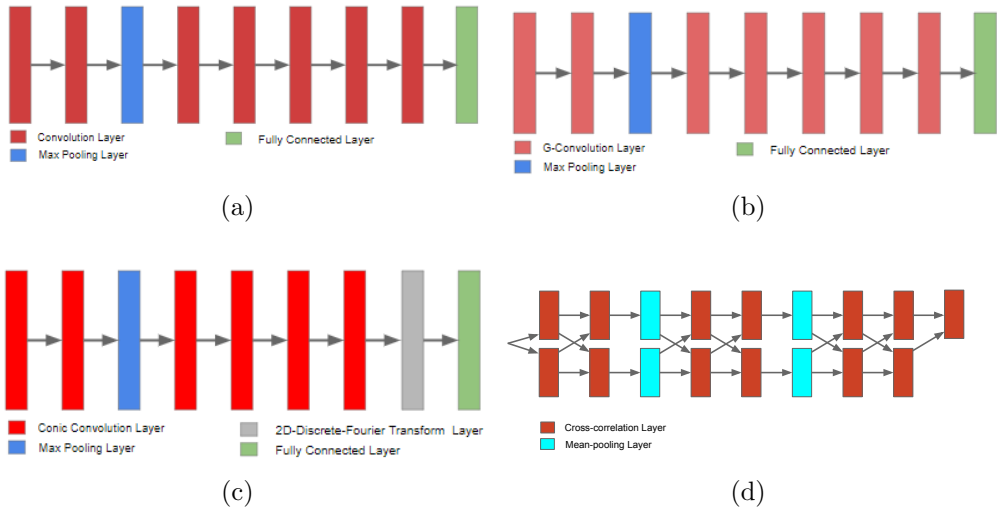


Figure 4: (a) Standard CNN (b) G-CNN (c) CFNet (d) H-Nets

3.1.5 Results

Here we present training and validation accuracy of the four models in Figure 5. All the models perform better on training dataset over validation, in other words, overfitting. Especially, standard CNN achieves 100% training accuracy while less than 80% validation accuracy. After fifty epochs, H-Nets get nearly the same training and validation accuracies. G-CNN converges fastest in less than 10 epochs while H-Nets converge slower taking almost fifty epochs.

Table 1 presents results on test dataset of rotated MNIST where H-Nets obtain the highest accuracy but taking more time than the three other models which is clearly shown in Figure 6. G-CNN gets nearly the same accuracy as H-Nets while eight minutes faster. CFNet obtains lower accuracy and slower than G-CNN. Standard CNN is the fastest model which only costs five minutes to finish training on rotated MNIST, however, the accuracy is not competitive with the other models.

In order to reduce the impact of other elements such as batch size, learning rate, data augmentation, we do not adopt the same network configurations as described in papers Cohen and Welling (2016), Worrall et al. (2016) and Chidester et al. (2019). For G-CNN, paper Cohen and Welling (2016) gets 2.28% test error,

97.72% test accuracy, which is higher than our result. The reason we think is they use Chainer and the last layer is a max pooling layer while we use Tensorflow and the last layer is a fully connected layer which introduces more parameters and thereby lowers the test accuracy. For CFNet, paper Chidester et al. (2019) reports 1.75% test error, 98.25% test accuracy. It introduces data augmentation based on code posted by the authors of G-CNNs at `https://github.com/tscohen/gconv_experiments`. However, the paper Cohen and Welling (2016) dose not state the use of training augmentation and the code does not use it by default. Therefore, the results of paper Chidester et al. (2019) are not very precise to use for comparison. For H-Net, paper Worrall et al. (2016) reports a test error of 1.69% and 98.31% accuracy. The gap between it with our result is mainly because the authors train with batch size 64 while we train with batch size 100. Although we scarify accuracy to make a more reasonable comparison, our results do prove that rotation equivariant and/or invariant CNNs improve image classification compared with standard CNNs.

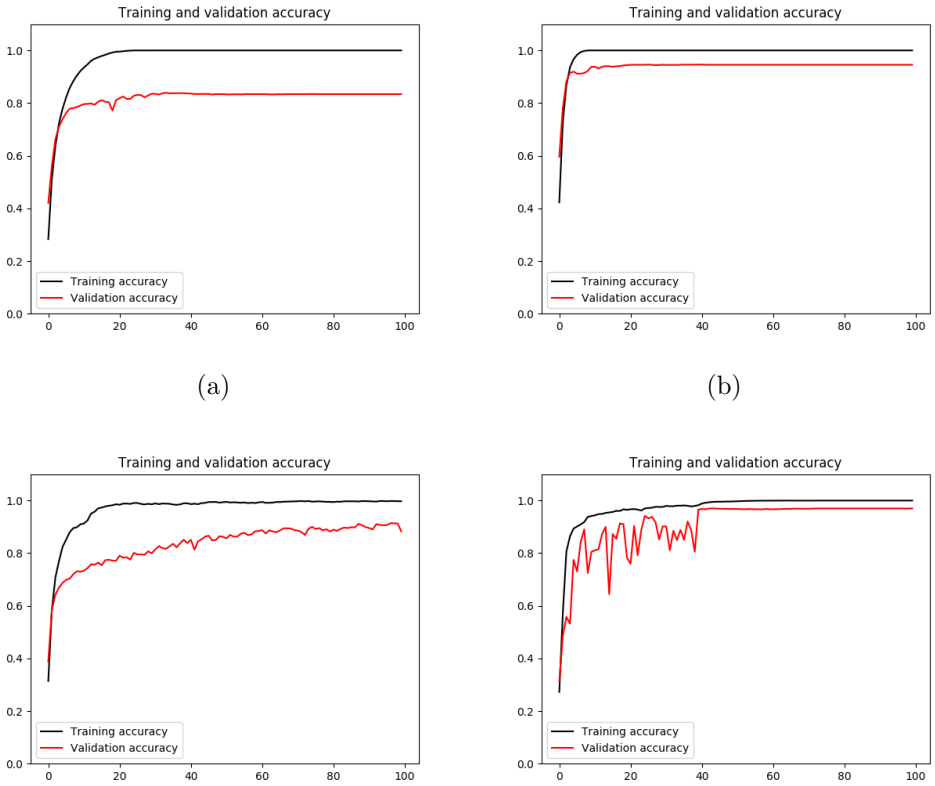


Figure 5: Training and validation accuracy of the four models on Rotated MNIST: (a) Standard CNN (b) G-CNN (c) CFNet (d) H-Nets

Table 1: Testing Results

Model	Accuracy	Time
H-Nets	96.98%	42min
G-CNN	95.35%	34min
CFNet	92.36%	40min
Standard CNN	76.19%	5min

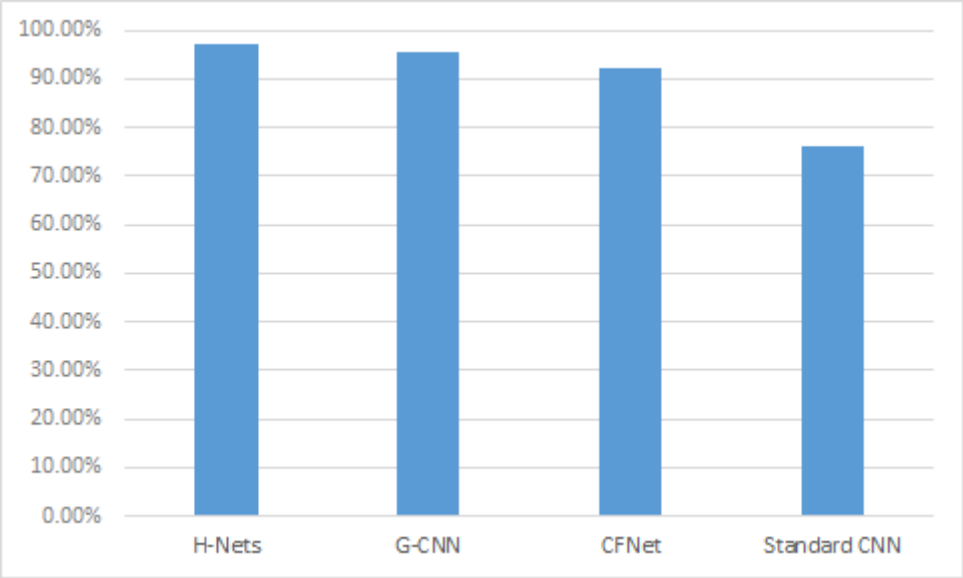


Figure 6: Test Accuracy of Standard CNN, CFNet, G-CNN and H-Nets on Rotated MNIST Dataset.

3.2 Oral Cancer

In the dataset there are 128,821 images with the size 80x80x3 where ∼33.2% are cancerous cells and the remaining healthy cells. Samples by 12 patients were collected at Folktandvården in Stockholms län by scraping the oral cavities. The samples were put in liquid vials, stained and scanned to digital images. The produced RGB images have been segmented to make this dataset where one image contains one cell each. The images are originally divided in a training and a test dataset as shown in Table 2. Different sizes of the training dataset are tested to evaluate the classification accuracy.

Table 2: Original dataset with oral cell images.

Training data		Test data	
Cancerous	Healthy	Cancerous	Healthy
22 454	50 851	20 323	35 193

We normalize this dataset by two different sets of mean and stand deviation. One is calculated over training set, while the other is calculated over individual image. We apply early stop feed back method which stops the training once no improvement during fifteen epochs in validation accuracy and select the model based on validation accuracy.

3.2.1 Standard CNN

We use Resnet50 (He et al., 2015) to evaluate Oral Cancer Dataset. Resnet50 is a residual learning framework to ease the training of networks that are substantially deeper than those used previously. It is proposed by Kaiming He and won the ImageNet competition in 2016. Keras provides multiple ready-made deep neural networks architecture. In this method, we adopt a ready-made ResNet50 model architecture but changing the input layer and the last layer.

3.2.2 G-CNN

We adopt the same configuration of G-CNN as described in Section 3.1.2 to evaluate Oral Cancer Dataset.

3.2.3 CFNet

Different from the architecture we use in rotated MNIST as described in Section 3.1.3, we adopt the architecture of paper (Chidester et al., 2019) which contains seven conic convolution layers, each has 32 3x3 filters, two average pooling layers, a 2D-DFT transition layer and one fully connected layer. The first three conic convolution layers contain 8 conic regions which are decreased to 4 in the following conic convolution layers and the 2D discrete Fourier transform transition layer. The average pooling layers have filter size 2x2 and stride 2. A dropout probability of 0.8 is applied at the final layer. The model is trained using learning rate 0.005, a wight decay l_2 penalty of 0.0005, batch size 50 and epochs 100.

3.2.4 H-Nets

The same configuration of H-Nets as described in Section 3.1.4 is applied when evaluating the method on Oral Cancer Dataset.

3.2.5 Results

The classification accuracy of two normalization methods with training sizes 10000, 30000 and 50000 predicted by models CFNet, G-CNN, H-Nets and Resnet50 are shown in Table 3 and Table 4. Table 3 shows the classification accuracies of trained models on dataset normalized by mean and standard deviation calculated over training set. With training size 10000 and 50000, H-Nets perform best while with training size 30000, Resnet50 beats H-Nets. All models get best results with training size 30000. Table 4 presents classification accuracies achieved on dataset normalized by mean and standard deviation calculated over individual image. Unlike Table 3, H-Nets perform best with training size 10000 and 30000,

while Resnet50 defeats H-Nets with training size 50000.

CFNet performs worse or slightly better than G-CNN in both Table 3 and Table 4. CFNet model used in Oral Cancer Dataset consists of 32 filters in every layer, which is much more than G-CNN's 10 filters in every layer. Additionally, from Figure 7, Figure 8, Figure 9 and Figure 10 we can see that CFNet is more sensitive to the two normalization methods, we conclude that CFNets are not very competitive with G-CNNs. The performance does not change monotonically with the increasing size of the training set, the reason might be that we choose the models that perform best on validation set to predict the test set. Small validation set implies statistical uncertainty around the estimated average test error and might not be representative for test set.

Table 3: Testing Results on Oral Cancer Dataset normalization by mean and standard deviation calculated by training set.

Training Size	H-Nets	CFNet	G-CNN	Resnet50
10000	**73.86**%	67.92%	72.16%	71.81%
30000	76.58%	73.88%	73.26%	**77.06**%
50000	**75.49**%	72.96%	72.88%	73.99%

Table 4: Testing Results on Oral Cancer Dataset of normalization by mean and standard deviation calculated by every image.

Training Size	H-Nets	CFNet	G-CNN	Resnet50
10000	**75.25**%	63.72%	72.35%	67.55%
30000	**76.01**%	61.13%	73.40%	72.03%
50000	75.18%	65.87%	71.55%	**76.03**%

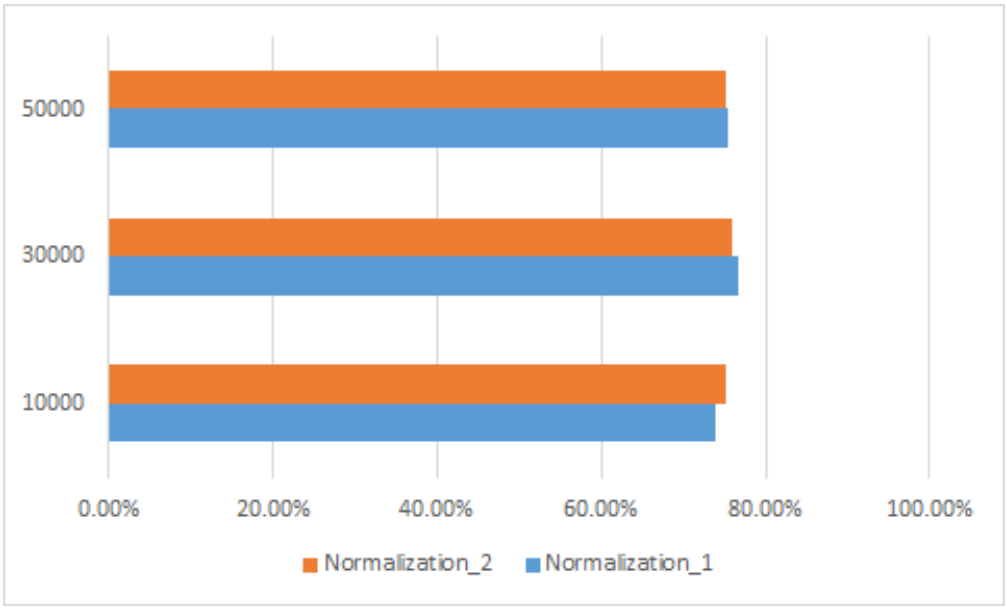


Figure 7: Classification accuracy on Oral Cancer Dataset of H-Nets with two normalization strategies used. normalization_1 denotes normalization by mean and standard deviation calculated by training set and normalization_2 is normalization by mean and standard deviation calculated by every image.

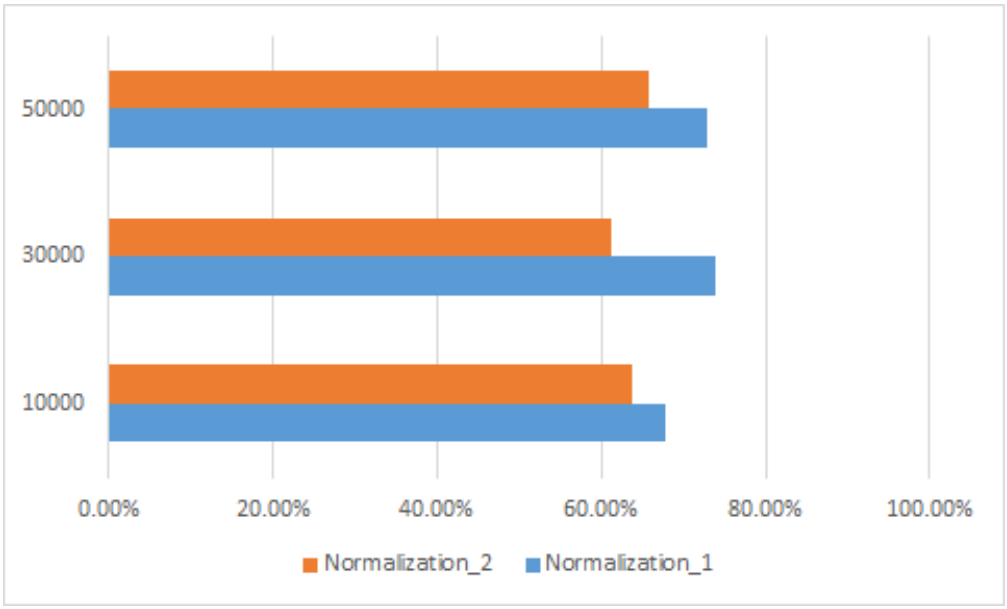


Figure 8: Classification accuracy on Oral Cancer Dataset of CFNet with two normalization strategies used. normalization_1 denotes normalization by mean and standard deviation calculated by training set and normalization_2 is normalization by mean and standard deviation calculated by every image.

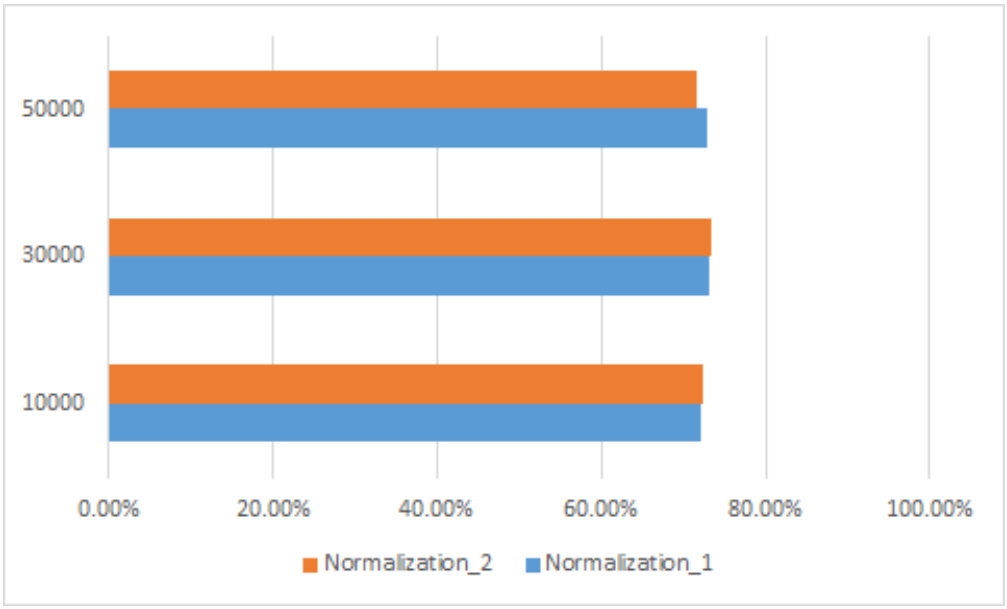


Figure 9: Classification accuracy on Oral Cancer Dataset of G-CNN with two normalization strategies used. normalization_1 denotes normalization by mean and standard deviation calculated by training set and normalization_2 is normalization by mean and standard deviation calculated by every image.

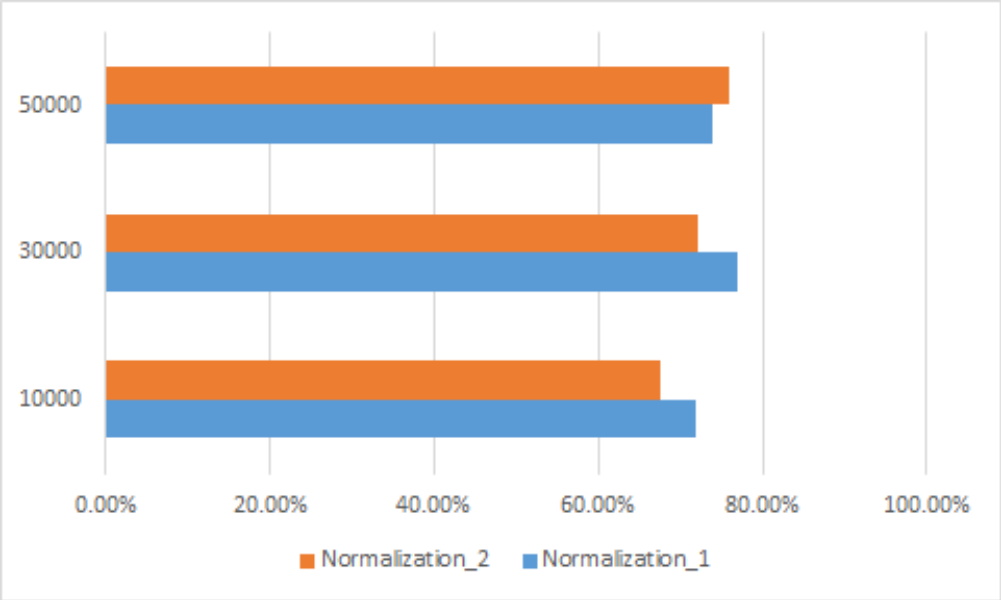


Figure 10: Classification accuracy on Oral Cancer Dataset of Resnet50 with two normalization strategies used. normalization_1 denotes normalization by mean and standard deviation calculated by training set and normalization_2 is normalization by mean and standard deviation calculated by every image.

4 Discussion and Future Work

The experiments presented here show that rotation equivariant and/or invariant CNNs do improve image classification with computational overhead. H-Nets exhibit good performance for both the rotated MNIST and the Oral Cancer Dataset and are not very sensitive to data pre-processing methods. However they require much more training time even with a simple architecture. G-CNNs are very competitive with H-Nets. They perform nearly as good as H-Nets but are much faster. With complex and huge datasets, the time saving might be significant. CFNets are supposed to be faster than G-CNNs meanwhile performing as good as G-CNNs. However, based on our experiments, with similar architecture, CFNets take more time and predict with less accuracy meanwhile they are very sensitive to pre-processing methods.

None of the models reached the current state of the art accuracy of 77.7%, but they all, and H-Net in particular, perform very well on Oral Cancer Dataset. The relatively low state-of-the-art accuracy may be caused by weak labeling of the dataset: the samples are labeled on the patient level, and not on the cell level. In reality, a patient with a cancer diagnosis probably also has healthy cells.

In the future, given time and resources such as high performance GPUs, we plan to apply the rotation equivariant and/or invariant methods to state-of-art CNN models such as Resnet, Lenet to explore if we can reach higher classification accuracy.

References

B. Chidester, T. Zhou, M. N. Do, and J. Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz353. URL https://doi.org/10.1093/bioinformatics/btz353.

T. S. Cohen and M. Welling. Group equivariant convolutional networks, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Lohrelei. Cat kitten silhouette free photo. https://www.needpix.com/photo/download/534886/cat-kitten-silhouette-black-shadow-outline-domestic-cat-mieze-felidae. [Online; accessed 10-January-2020].

W. H. Organization. Oral cancer. https://www.who.int/cancer/prevention/diagnosis-screening/oral-cancer/en/. [Online; accessed 12-December-2019].

public_static_twiki. Variations on the MNIST digits.

B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant CNNs for digital pathology. June 2018.

D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *CoRR*, abs/1612.04642, 2016. URL http://arxiv.org/abs/1612.04642.

D. Öhman. Oral cancer och maligna tumörtecken, 2019.