Feature Map Upscaling to Improve Scale Invariance in Convolutional Neural Networks

Dinesh Kumar[1][a] and Dharmendra Sharma[2][b]
[1]Faculty of Science & Technology, University of the South Pacific, Laucala Bay Road, Suva, Fiji
[2]Faculty of Science & Technology, University of Canberra, 11 Kirinari Street, Canberra, ACT 2617, Australia

Keywords: Convolutional Neural Network, Feature Map, Filter Pyramid, Global Feature, Scale Invariance, Visual System.

Abstract: Efforts made by computer scientists to model the visual system has resulted in various techniques from which the most notable has been the Convolutional Neural Network (CNN). Whilst the ability to recognise an object in various scales is a trivial task for the human visual system, it remains a challenge for CNNs to achieve the same behaviour. Recent physiological studies reveal the visual system uses *global-first* response strategy in its recognition function, that is the visual system processes a wider area from a scene for its recognition function. This theory provides the potential for using global features to solve transformation invariance problems in CNNs. In this paper, we use this theory to propose a *global-first* feature extraction model called Stacked Filter CNN (SFCNN) to improve scale-invariant classification of images. In SFCNN, to extract features from spatially larger areas of the target image, we develop a trainable feature extraction layer called Stacked Filter Convolutions (SFC). We achieve this by creating a convolution layer with a pyramid of stacked filters of different sizes. When convolved with an input image the outputs are feature maps of different scales which are then *upsampled* and used as global features. Our results show that by integrating the SFC layer within a CNN structure, the network outperforms traditional CNN on classification of scaled color images. Experiments using benchmark datasets indicate potential effectiveness of our model towards improving scale invariance in CNN networks.

1 INTRODUCTION

Understanding a scene from just a single exposure is a trivial task for the visual system (Han et al., 2017), for example being able to recognise an object in various scales (scaling), able to distinguish rotated objects such as the ability to read signs on the wall while in laying position (rotation) or identify a moving object (translation). Efforts by computer scientists to model this behaviour has resulted in various techniques from which most notable in the last decade has been the Convolutional Neural Network (CNN) (LeCun et al., 1998). CNNs have achieved great success in numerous computer vision tasks such as in image classification, object detection and recognition, semantic segmentation and boundary detection.

The short comings of CNNs however, are in its inability to adequately handle invariances introduced in similar images it has been trained on (Jaderberg et al., 2015; Kauderer-Abrams, 2017; Lenc and Vedaldi, 2015). Invariance refers to the ability of recognising objects even when the appearance varies in some ways as a result of transformations such as translations, scaling, rotation or reflection. Recent physiological studies of the visual pathway reveal the visual system uses both local and global features in its recognition function. Cells tuned to global features respond to visual stimuli prior to cells tuned on local features leading to suggestions of a *global-first* response strategy of the visual system to speed-up recognition (Huang et al., 2017; Su et al., 2009). This theory provides the potential for using global features combined with local features to solve transformation invariance problems in CNNs.

In this paper, we address improving scale-invariant classification in CNNs by exploiting the *global-first* theory and propose a trainable feature extraction layer called Stacked Filters Convolution (SFC). The design of SFC layer enables the network to extract global features extracted from spatially larger areas of the target image. Inspired by the work of (Peng et al., 2017), we apply the concept of

[a] https://orcid.org/0000-0003-4693-0097
[b] https://orcid.org/0000-0002-9856-4685

Kumar, D. and Sharma, D.
Feature Map Upscaling to Improve Scale Invariance in Convolutional Neural Networks.
DOI: 10.5220/0010246001130122
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 113-122
ISBN: 978-989-758-488-6

large filters (kernels) to cover broader areas of an image during convolution operation. In SFC layer, we create pyramids of stacked filters of different sizes. The stacking of filters create a conical filter structure referred to as filter pyramids. Input image data passing through SFC layer are convolved with each filter resulting in feature maps of different scales. These feature maps are then upscaled and used as global features and passed to the next layer in the CNN network. In this work, we refer to the integration of SFC layer within an existing CNN as Stacked Filter CNN (SFCNN) model.

We conduct extensive experiments to evaluate scale invariance performance of SFCNN. We use LeNet5 CNN (LeCun et al., 1998) as our benchmark model. First, the datasets are trained on LeNet5 to establish benchmark results for comparison. Then we ensemble LeNet5 with SFC layer by placing it as the first layer in LeNet5 feature extraction pipeline. This location enables the SFC layer to pass features extracted from spatially broader areas of the image (*global-first*) into the CNN network for further processing. We train the ensemble SFCNN on the same datasets. We study the performance of SFCNN in classifying image samples on specific scale categories. We also study the performance of SFCNN on individual classes where images from various scales per class are evaluated. For consistency we use the same test samples on all models developed. In all our case studies, performance of SFCNN are compared with our benchmarks. Our results show SFCNN outperforms traditional LeNet5 CNN in classifying color images across majority of the scale categories. In addition, we report promising results on SFCNN's ability to classify images in various scale levels for each dataset class in particular for color images.

The main contributions of this paper are to improve CNNs towards classification of scaled images by showing the effectiveness of a) processing spatially broader areas of an input image in the initial stages of a CNN feature extraction pipeline, and b) enhancing features extracted by applying upscaling on feature maps.

The rest of the paper is organised as follows: Section 2 reviews related work while Section 3 introduces our model. Section 4 describes our experiment design and results are presented in Section 5. We summarise and point to future directions in Section 6.

2 BACKGROUND

Use of Global Features in CNNs. While local features are effectively extracted in CNNs using small filters performing a patch-wise operation with the target image, extracting global features requires studying the whole image or spatially larger areas of the target image. Here, local features are classified as lines (edges) and curves while shape, colour and shape contours are labelled as global features. In some studies, global features have been studied and applied in CNNs but are limited to using feature descriptors such as histogram of gradients (HOG) (Zhang et al., 2016) and SIFT (Zheng et al., 2017). However, they have not been tested on the networks ability to be spatially invariant and also feature extractors such as HOG and SIFT are non-trainable.

Use of Large Kernels in CNNs. The use of large kernels to extract features from spatially broader areas of the target image have been studied in some work. In the area of semantic segmentation (Peng et al., 2017) proposed a Global Convolutional Neural Network in which they studied the use of large kernels. Instead of directly applying large kernels as normal convolutions, they used a combination of vector type kernels of size $1 \times k + k \times 1$ to connect with a large $k \times k$ region in the feature map. They conducted their experiments on PASCAL VOC dataset and concluded that large kernels play an important role in both classification and localisation tasks. In their design, they did not use any non-linearity after convolution layers as is the practice in standard CNN models. In another piece of work, (Park and Lee, 2016) inform extracting information from a large area surrounding the target pixel is essential to extract for example texture information.

Pyramid based Methods in CNNs for Scale-Invariant Classification. Pyramid based methods have been used to address scale invariance in CNNs to some extent but have been limited to either generating image pyramids or feature map pyramids. For example, (Kanazawa et al., 2014) describe work where they first create an image pyramid by scaling the target image and using the same filter to convolve all scaled input. The feature maps generated are normalised to obtain the same spatial dimensions and then pooled to obtain a locally scale-invariant representation. However, in their implementation, scaling the target image is similar to applying scale augmentation. In our work, we present no augmentation of the input images. In another work, (Xu et al., 2014) propose a scale-invariant CNN (SiCNN) by applying a similar process of convolving a filter on different image scales. (Lin et al., 2017) exploit the pyramidal hierarchy of feature maps in deep convolutional networks by developing lateral connections from each

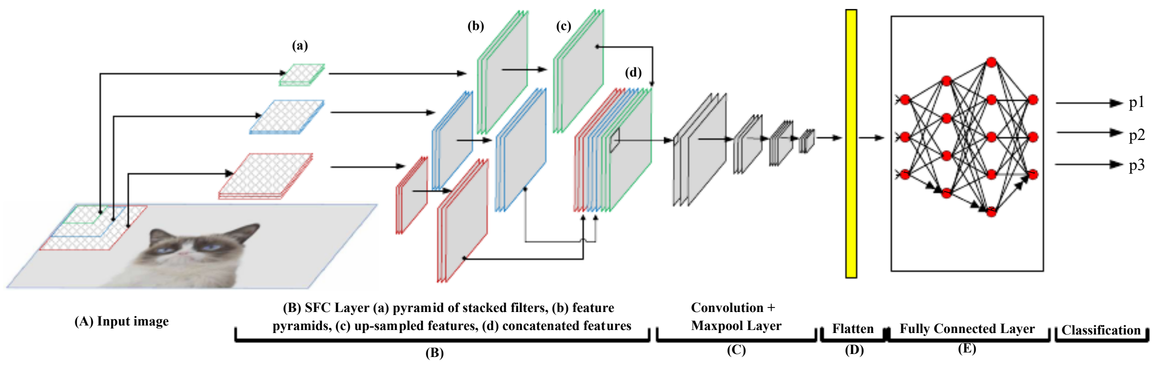
(a)
(b)
(c)
(d)
p1
p2
p3
(A) Input image
(B) SFC Layer (a) pyramid of stacked filters, (b) feature pyramids, (c) up-sampled features, (d) concatenated features
(B)
Convolution + Maxpool Layer
(C)
Flatten
(D)
Fully Connected Layer
(E)
Classification

Figure 1: Architecture of SFCNN with $k = 3$ filters in SFC layer. An input image (A) is passed to the SFC layer (B) which applies convolutions with each filter from the filter stack (a) producing f_{1-3} output feature maps of different scales (b). The output feature maps are then upsampled to generate uniform-sized outputs (c). The upscaled features are concatenated and passed to the CNN network (C) for further feature extraction. The flatten layer (D) vectorises the final output from the CNN network and forwards to the classifier for learning (D).

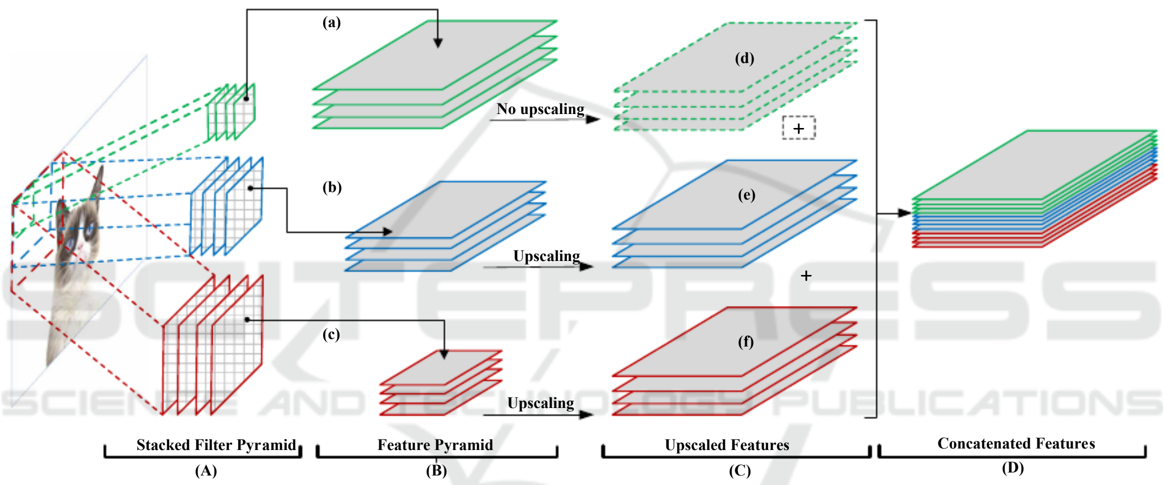

Figure 2: A detailed lateral view of the SFC layer with $k = 3$ filter stacks (A). Each stack in the filter pyramid contains further n-filters. An input image is convolved using each filter from the filter stack producing k stacks of n output feature maps of different sizes (B)-((a),(b),(c)). The feature maps are then upsampled to generate a uniform sized feature outputs (C)-((d),(e),(f)). The upscaled features are finally concatenated (D) for forward propagation.

feature map in the pyramid to build high-level semantic feature maps at all scales. They show that feature pyramids generated in this way are scale-invariant as a change in an object's scale is offset by shifting its level in the pyramid. Similar architectures are proposed in works of (Kim et al., 2018; Zhao et al., 2019; Kong et al., 2018).

Based on the concept of large kernels and pyramid based methods, (Kumar and Sharma, 2020) propose a distributed information integration CNN model called D-Net by combining local and global features from images. To extract global features, they developed a trainable layer called Filter Pyramid Convolutions (FPC). In FPC layer, various scale filters (from small to large filters) are applied to progressively cover broader areas of an image. The features extracted are then pooled, resulting compact sized feature maps as output in terms of its spatial dimension. This design limited the output of the FPC layer to be used as input in subsequent convolution layers. In this paper, we adopt a similar design as the FPC layer for our SFC layer. However, to overcome the problem of small scaled output feature maps from FPC layer, the feature maps in SFC layer are instead upscaled. The upscaled feature maps allows SFC layer output to be used as input in subsequent feature extraction layers of a CNN.

Whilst much progress and state-of-the-art results are shown there is still little research that show the effectiveness of exposing a CNN with global view of input images to solve scale-invariant classification problem in CNNs. This paper achieves to fill this gap.

3 MODEL

In this section we propose a novel method called Stacked Filters Convolution (SFC) that allows a CNN network take advantage of large kernels to extract global features to improve classification of scaled images. The design of SFC layer is inspired by the work of (Huang et al., 2017) who show that biological visual system utilizes global features prior to local features in detection and recognition. The integration of SFC layer within an existing CNN is referred to as Stacked Filter CNN (SFCNN). The SFCNN model comprises of five main parts (A-E) as shown in Figure 1. They are explained in the following sub-sections.

3.1 Stacked Filters Convolution (SFC) Layer

Unlike in a traditional CNN where normally a small filter is applied for each convolution in a convolution layer, the SFC layer houses a battery of filters of varying sizes. This forms a pyramidal structure of stacked filters and operates on the target image using the same standard sliding window convolution technique (Figure 1(a)). The important part in the setup of SFC is determining the sizes of each filter in the stack. Here, the dimensions (k_h, k_w) of each filter in the stack is manually chosen. The size of the largest filter (base of the pyramid) (k_h^0, k_w^0) is carefully chosen so as to allow the convolution to produce a 2D output map (f_h^0, f_w^0) (considering height and width only). It is required that the output feature map size (f_h, f_w) from each filter is calculated after considering the other hyper parameters *stride* and *padding*. The size of the next filter (k_h^1, k_w^1) is chosen by determining the output size of its resultant feature map (f_h^1, f_w^1) where f_h^1 and f_w^1 is a multiple of f_h^0 and f_w^0 respectively. This procedure is required in order to allow upscaling of feature map (f_h^0, f_w^0) by an integer upscaling factor. Subsequently sizes of other filters are identified using a similar process.

For example, for a 32×32 image, the filter sizes chosen for the stack is 25×25, 17×17 and 3×3 with upscaling factors 4, 2 and 1 respectively. Table 1 describes how the final output size of the SFC layer is calculated after considering the filter sizes, upscaling factors and appropriate uses of stride and padding on the target image.

Since each filter produces a different size feature map (Figure 1(b)), these maps then need to be normalised to produce a uniform size final feature maps. Here, we use the technique of upscaling and in our work use the *bilinear* upscaling method. The smaller feature maps are upscaled using a scale factor to produce an output equal to the largest feature map (Figure 1(c)). The largest feature map is unchanged. Applying this approach to the above example results in the final output size as shown in Table 1. Finally all upsampled feature maps are concatenated and passed to the next layer (Figure 1(d)). In our implementation since the SFC layer is the only layer that gets to inspect the target image, we maintain the inclusion of a small filter in our stack. This is done to allow extracting local features from the target image which we would otherwise lose. In this way, SFC layer also allows local features to be collected and packed together with global features for onward processing. Figure 2 shows the lateral view of the components of SFC layer and flow of information.

3.2 Forward Propagation Process in SFC

To achieve the forward pass, an input image is passed to the SFC layer which applies convolutions with each filter from a filter stack and outputs a stack of feature maps as a result. This process repeats for all stacks of filters in the layer resulting in stacks of output feature maps of different scales accordingly. The stack of output feature maps are then upsampled to generate uniform-sized outputs in terms of height and width of the feature maps in all stacks. The upscaled stack of features are finally concatenated for forward propagation into the network. The shape of each stack of upscaled features is saved for use in backward propagation. Since the remainder of the network is composed of traditional convolution, relu, max pooling, flatten and fully connected neural network layers, the forward propagation is as described in (LeCun et al., 1998).

3.3 Backward Propagation Process in SFC

The backward function in SFC layer receives gradients from the network. It then unstacks or slices the gradients in the exact same dimensions and shape of the individual stack of feature maps that were concatenated during the forward pass. This results in stacks of gradients maps corresponding to each stacked upscaled feature maps (during forward pass). Each stack of gradient is max pooled by the same factor that was used to upscale the feature map to reduce the dimensions of the feature maps. Using chain rule derivative algorithm these gradients are then used to update the weights of filters in the corresponding filter stacks.

Table 1: Calculation of final output feature map sizes in the SFC layer. The output size of the intermediate feature map in column (E) is dependent on the filter size (column (B)), *stride* and *padding*. It must also be a multiple of the sizes of other output maps in column (E).

(A) **image size** (I_h, I_w)	**(B)** **filter size** k_h, k_w	**(C)** **stride** s	**(D)** **padding** p	**(E)** **feature map output size** (f_h, f_w) $(f_h = (I_h + 2p) - k_h + 1)/s$ $(f_w = (I_w + 2p) - k_w + 1)/s$	**(F)** **up-scaling factor**	**(G)** **final output map size** (O_h, O_w)
32×32	25×25	1	0	8×8	4	32×32
32×32	17×17	1	0	16×16	2	32×32
32×32	3×3	1	1	32×32	1	32×32

4 THE EXPERIMENTS

We describe the datasets, CNN architecture, SFC parameters and our experimental design in the following sub sections.

4.1 Dataset Description

Fashion-MNIST: The Fashion-MNIST (FMNIST) dataset (Xiao et al., 2017) consists of 60,000 training images and 10,000 test images of fashion products from 10 categories. The sample images are greyscale (1-channel) of size 28×28 pixels. The training and test batches have equal distribution of the number of samples from each class.

CIFAR10. CIFAR10 dataset (Krizhevsky et al., 2009) consists of 60,000 colour images of size 32×32 pixels with 3-channels. The dataset is divided into 50,000 training and 10,000 test samples. The samples are divided into 10 mutually exclusive classes defining various objects. The train and test batch contain equal number of images from each class.

4.2 CNN Architecture and SFC Parameters

For benchmarking and also for combining global features through SFC to a CNN network we used LeNet5 CNN structure as described below.

LeNet5 Network. Proposed by (LeCun et al., 1998), the LeNet5 network in our work comprises of three sets of convolution layers and two max pooling layers. The architecture is described in Table 2. Since we are using two datasets with different dimensions for the input images (32×32 for CIFAR10 and 28×28 for FMNIST) the hyper-parameter *padding* for the second convolutional layer in the LeNet5 network trained on CIFAR10 is set to 1. For LeNet5 model trained on FMNIST, *padding* for the first and second convolutional layers is set to 2 and 1 respectively.

SFC Parameters. For training on CIFAR10 dataset we setup SFC layer with 3 stacks of filters ($k\_stacks = 3$) having filters of sizes (3×3), (17×17) and (25×25) respectively. Each stack is initialised with 6 filters ($n\_filters = 6$). We set $stride = 1$ for all stacks, $padding = 1$ for stack with 3×3 filters, $padding = 0$ for the rest of the stacks and upscaling factors $4, 2, 1$ respectively for each filter stack. The final shape of the concatenated stacks of feature maps on CIFAR10 dataset is ($18 \times 32 \times 32$) where $18 = k\_stacks \times n\_filters$. On FMNIST dataset, we use the same values for parameters in SFC except the filter sizes in each stack are changed to (3×3), (15×15) and (22×22). The final shape of the concatenated stacks of feature maps on FMNIST dataset is ($18 \times 28 \times 28$).

4.3 Training Process

First we train the benchmark CNN (LeNet5) on CIFAR10 and FMNIST datasets separately. This establishes our benchmark results against which we compare results of SFCNN networks. Then we integrate SFC layer within LeNet5 network pipeline as described in Figure 1. We train SFCNN using the same training parameters as used on LeNet5 resulting in SFCNN models for CIFAR10 and FMNIST datasets respectively. Hence, we obtain a total of four trained models for comparison (two models per dataset).

End-to-end training was performed on all models. For networks trained on CIFAR10 dataset we start with a warm-up strategy for 4 epochs with a learning rate of 10^{-2}, 10^{-3} from epochs 5-50 and decreasing it to 10^{-4} for the rest of training. For training on FMNIST dataset the learning was adjusted to 10^{-2} for 2 epochs, 10^{-3} from epochs 3-50 and decreasing it to

10^{-4} for the rest of training. Training on all models were stopped at 100 epochs. Stochastic gradient decent and cross-entropy were used as learning and loss function respectively. We use weight decay of 10^{-4} and momentum of 0.9. For training we use batch size of 8 and 4 for testing. We implemented our models using PyTorch version 1.2.0 on a Dell Optiplex i5 48GB RAM computer with Cuda support using NVIDIA GeForce GTX 1050 Ti 4GB graphics card.

4.4 Preparing Scaled Images for Testing Models

To test our models on scaled images we first establish 7 scale categories - $[150, 140, 120, 100, 80, 60, 50]$. The numbers indicate percentage an image is scaled to. In this research we consider both reduction and enlargement of image size from the original. We select at random 100 images per class from CIFAR10 and FMNIST test datasets. Then, we scale each image to a size defined in our scale category list. Since the images in our dataset are small we stop at scale 50%. In this fashion for a single test image of a class we generate 7 scaled test images. Each scaled image is allocated to its own class and scale category folder resulting in 1000 scaled images in each of the 7 scaled categories. We further pool all images from all 7 scale categories into an ensemble test dataset resulting in 7000 scaled images. We analyse our models on scaled images from each of these scale categories independently (Section 5.2). Finally we use the ensemble dataset to analyse the performance of our models for individual classes (Section 5.3). Figure 3 shows an example image from each dataset and its corresponding scaled versions for testing. Table 3 provides detailed information on the number of scaled images generated for testing.

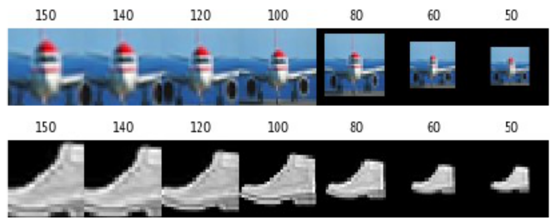

Figure 3: An example of scaled test images from datasets CIFAR10 - airplane (*top*) and FMNIST - ankle boot (*bottom*). The numbers indicate percentage image is scaled to. 100 indicates no scaling.

4.5 Evaluation Metrics

Analysis on Scale Categories. We use metrics *precision*, *recall* and *accuracy* to analyse results of SFCNN on scale categories. Since we do not have imbalanced class distribution in our datasets we use macro-average weighting to calculate *precision* and *recall*.

Analysis on Dataset Classes. We use metrics *sensitivity* (recall) and *specificity* to analyse results of SFCNN for scaled images in individual dataset classes.

5 RESULTS AND DISCUSSION

5.1 Comparing Model Training Statistics on Regular Images

Table 4 compares the train losses and test accuracy for all the networks used in our experiments on regular images from the test datasets. These are evaluations on images that have not been subjected to any form of scale transformations. Our ensemble SFCNN model outperforms the traditional LeNet5 network on test accuracy metrics (indicated in bold). The highest test accuracy increase of 2.3% is recorded on SFCNN on CIFAR10 dataset. This indicates combining global feature information in network training is useful in improving the overall generalisation capability of the models, in particular for color images.

5.2 Effects of Feature Map Upscaling on Classification of Scaled Images

The classification results of our models on different scale categories and on different datasets can be viewed in Tables 5 and 6. From these results, we arrive at three observations.

First, classification accuracies on CIFAR10 scaled categories obtained by SFCNN networks show the inclusion of the SFC layer as a global feature extractor, gives promising results in classification of scaled imaged. SFC provides significant improvement in the overall network's ability to classify scaled images compared to the benchmark LeNet5 network. The column *hit-rate* in Tables 5 and 6 indicate the number of scale categories SFCNN outperformed the benchmark. For purposes of our study hit-rate of $>= 50\%$ is desirable, that is SFCNN should at least perform better on 50% of the scale categories compared to the benchmark LeNet5 only network. Since the ensemble

Table 2: Architecture of LeNet5 network used in our experiments (Hosseini et al., 2017).

Model	**Layers**
LeNet5	(conv 5x5x6) $\rightarrow$ (maxpool 2x2) $\rightarrow$ (conv 5x5x16) $\rightarrow$ (maxpool 2x2) $\rightarrow$ (conv 5x5x120) $\rightarrow$ (fc 84) $\rightarrow$ (fc 10) $\rightarrow$ softmax

Table 3: Information on scaled images generated for testing our models.

Number of classes	10
Scale categories	150,140,120,100, 80,60,50
Image per class	100
Total images per category	1000
Total ensemble images over all scale categories	7000

test dataset combines all scaled images in one batch it is excluded from this ratio. The hit-rate on CIFAR10 accuracy scores are $> 70\%$ which is well above the desired threshold meaning the model was able to identify a higher number of samples in its correct class despite the images being scale transformed. Since we apply bilinear upscaling of feature maps in SFC layer which enlarges the extracted features, we test the effects of this operation particularly on the scale reduced images (categories 80, 60 and 50 in Tables 5 and 6). We observe SFCNN on overall performs better than LeNet5 on these scale categories, specifically on scale category 80 where the test accuracy is higher by 8.6%. Performance on SFCNN on FMNIST scale categories however are not very promising where LeNet5 benchmark results are higher. This indicates SFCNN works better on color images than on grey-scale images.

Second, macro-average precision reveals SFCNNs ability to classify a high number of scaled images identified as positive to be actually positive. On CIFAR10 dataset, SFCNN achieves a hit-rate of 100% over all scale categories. On the contrary we observe the benchmark LeNet5 model performing better on FMNIST dataset compared to on CIFAR10. These results further show SFCNNs ability to classify color scaled images better but at the same time its inability to achieve a similar performance on grey-scale images. We also note in general precision scores of both SFCNN and the benchmark LeNet5 models decline with increasing scale reduction (Figure 4).

Third, macro-average recall statistics of all models on both datasets are identical to the accuracy scores. While this shows SFCNNs superior ability to return most of the relevant results in nearly all scale categories compared to the benchmark LeNet5 on CIFAR10 dataset, LeNet5 on the other hand has higher recall scores on FMNIST grey-scale images.

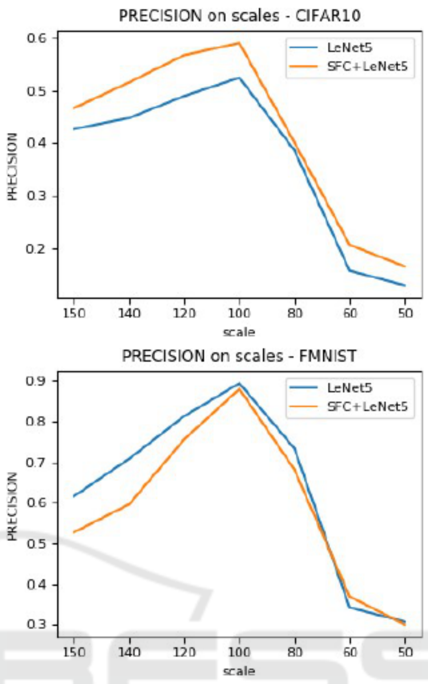

Figure 4: Drop in precision with declining scale of images.

5.3 Performance of SFCNN on Individual Classes

The classification results of the studied models on ensemble scaled test set evaluated on datasets classes can be viewed in Tables 7 and 8. From these results, we arrive at two observations.

First, sensitivity (recall) scores on SFCNN network have reasonable hit-rate (50%) showing better performance on most classes than the benchmark LeNet5 models on CIFAR10. This is however not consistent on both datasets where the hit-rate on sensitivity score on FMNIST dataset is below our desired 50% threshold. Here $hit-rate$ is the ratio of counting the number of classes SFCNN produced a higher sensitivity score than the benchmark LeNet5 to the total number of classes. Though comparatively achieving higher sensitivity scores than the benchmarks on CIFAR10 dataset, we note that the scores for majority of the classes in both datasets are low (below 50%). We investigated the false negative (FN) and true positive (TP) scores to reveal FN scores to be higher than TP for several classes but not in all. This is also true on the benchmark LeNet5.

Table 4: Train losses and test accuracy for all models used in our experiments.

Model	train loss CIFAR10	test acc CIFAR10	difference	train loss FMNIST	test acc FMNIST	difference
LeNet5	1.734	0.568		**1.535**	0.899	
SFCNN	**1.733**	**0.591**	**-0.001 (loss)** **+2.3% (acc)**	1.566	**0.901**	**+0.031 (loss)** **+0.2% (acc)**

Table 5: Performance summarization of the studied models on all the scale categories on CIFAR10 dataset.

		scale categories								
Model	**metric**	**ensemble**	**150**	**140**	**120**	**100**	**80**	**60**	**50**	**hit rate**
LeNet5	acc	0.381	0.449	0.478	0.531	0.577	0.265	0.217	0.149	0.714
SFCNN		**0.394**	0.438	**0.487**	**0.547**	**0.586**	**0.351**	0.193	**0.159**	(5/7)
LeNet5	precision	0.388	0.427	0.448	0.490	0.525	0.386	0.157	0.129	1.000
SFCNN		**0.419**	**0.467**	**0.516**	**0.568**	**0.591**	**0.401**	**0.206**	**0.165**	(7/7)
LeNet5	recall	0.381	0.449	0.478	0.531	0.577	0.265	0.217	0.149	0.714
SFCNN		**0.394**	0.438	**0.487**	**0.547**	**0.586**	**0.351**	0.193	**0.159**	(5/7)

Second, our tests results show specificity scores on overall are higher than sensitivity scores for all classes on both datasets and for each of the tested models. This is also true on the benchmark LeNet5 network. We investigated the true negative (TN) and false positive (FP) scores to reveal TN scores to be higher than FP for all classes. These results are promising as high TN scores indicates SFCNN is able to produce a high number of correctly predicted negative values. Comparatively SFCNN network on CIFAR10 performed better on specificity than the same networks on FMNIST dataset when considering the hit-rate indicating better performance on color than grey-scale images.

5.4 Model Complexity

The advantage of SFCNN lies in the application of larger kernels that are used to detect features from spatially larger areas of the input image. This overcomes the shortcomings of standard CNNs that usually address a small area of the input image or feature map at a time using smaller kernels. However, the limitations of SFCNN include a) an increase in network parameters due to the use of large kernels in the SFC layer, b) the SFC layer generates larger feature maps as a result of upscaling which in turn requires more convolution operations in the network, c) increase in the volume of feature maps due to concatenation of upscaled features maps in the SFC layer, and d) the model is still not able to inspect the entire image globally due to design constrains of the kernels in the filter pyramid. Limitations a), b) and c) further lead to increased computation time.

6 CONCLUSION

In this work we propose a method to learn scale invariance in CNNs by introducing a new technique of using large kernels to extract spatial information from the target image and combine it with local features for learning by the network. The proposed method called Stacked Filters Convolution (SFC) uses a stack of large and small filters arranged in a pyramidal structure. Each group of filters from the stack is convolved with the input image producing feature maps of different scales. These feature maps are upscaled to produce a uniform sized output map which is then passed to the next layer.

We study the effects of *global-first* feature extraction by adding the SFC layer and evaluating the networks ability to classify test images subjected to scale transformations and compare with our benchmark. Our results show overall improvements in classification of scaled images in comparison to classification results from our benchmark networks. Further, the results also indicate better performance of ensemble SFCNN network on color images than on grey-scale images. From our experimental results we conclude that spatial features extracted from larger areas of the target image during training help in improving the scale invariance capability of CNN networks, in particular for color images.

Problems and opportunities that require further investigations are to evaluate other upscaling methods as information may be lost due to the type of interpolation method used, test this technique to evaluate other forms of transformations such as rotations and translations and apply SFC layer with other bench-

Table 6: Performance summarization of the studied models on all the scale categories on FMNIST dataset.

		scale categories								
Model	**metric**	**ensemble**	**150**	**140**	**120**	**100**	**80**	**60**	**50**	**hit rate**
LeNet5	acc	0.611	0.575	0.654	0.785	0.895	0.703	0.373	0.295	0.000
SFCNN		0.566	0.480	0.573	0.743	0.880	0.647	0.369	0.267	(0/7)
LeNet5	precision	0.676	0.616	0.708	0.813	0.894	0.733	0.342	0.307	0.142
SFCNN		0.579	0.527	0.596	0.756	0.880	0.681	**0.369**	0.300	(1/7)
LeNet5	recall	0.611	0.575	0.654	0.785	0.895	0.703	0.373	0.295	0.000
SFCNN		0.566	0.480	0.573	0.743	0.880	0.647	0.369	0.267	(0/7)

Table 7: Performance summarization of the studied models on CIFAR10 classes.

	[CIFAR10 classes] [Scale category - ensemble]										
Model	**air-plane**	**auto-mobi le**	**bird**	**cat**	**deer**	**dog**	**frog**	**horse**	**ship**	**truck**	**hit rate**
metric	**- sensitivity scores -**										
LeNet5	0.353	0.469	0.000	0.377	0.316	0.439	0.624	0.339	0.430	0.463	
SFCNN	**0.389**	**0.517**	**0.240**	**0.469**	0.273	0.374	0.559	0.317	**0.439**	0.379	0.500 (5/10)
metric	**- specificity scores -**										
LeNet5	0.970	0.972	1.000	0.854	0.957	0.838	0.887	0.943	0.962	0.929	
SFCNN	0.959	0.946	0.950	**0.856**	**0.974**	**0.890**	**0.920**	**0.961**	0.926	**0.946**	0.600 (6/10)

Table 8: Performance summarization of the studied models on FMNIST classes.

	[FMNIST classes] [Scale category - ensemble]										
Model	**t-shirt -top**	**tro-user**	**pull-over**	**dress**	**coat**	**san-dal**	**shirt**	**snea-ker**	**bag**	**ankle-boot**	**hit rate**
metric	**- sensitivity scores -**										
LeNet5	0.796	0.824	0.424	0.521	0.303	0.933	0.206	0.510	0.926	0.671	
SFCNN	0.531	0.734	0.303	**0.727**	**0.313**	0.857	0.179	**0.561**	0.750	**0.700**	0.400 (4/10)
metric	**- specificity scores -**										
LeNet5	0.851	0.999	0.957	0.988	0.981	0.894	0.974	0.993	0.936	0.994	
SFCNN	**0.939**	0.991	**0.960**	0.901	0.970	**0.932**	0.966	0.968	0.900	0.990	0.300 (3/10)

mark network configurations using larger and more complex datasets.

REFERENCES

Han, Y., Roig, G., Geiger, G., and Poggio, T. A. (2017). Is the human visual system invariant to translation and scale? In *2017 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 27-29, 2017*.

Hosseini, H., Xiao, B., Jaiswal, M., and Poovendran, R. (2017). On the limitation of convolutional neural networks in recognizing negative images. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 352–358. IEEE.

Huang, J., Yang, Y., Zhou, K., Zhao, X., Zhou, Q., Zhu, H., Yang, Y., Zhang, C., Zhou, Y., and Zhou, W. (2017). Rapid processing of a global feature in the on visual pathways of behaving monkeys. *Frontiers in Neuroscience*, 11:474.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc.

Kanazawa, A., Sharma, A., and Jacobs, D. W. (2014). Locally scale-invariant convolutional neural networks. *CoRR*, abs/1412.5104.

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

Kim, S.-W., Kook, H.-K., Sun, J.-Y., Kang, M.-C., and Ko, S.-J. (2018). Parallel feature pyramid network for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 234–250.

Kong, T., Sun, F., Tan, C., Liu, H., and Huang, W. (2018). Deep feature pyramid reconfiguration for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

Kumar, D. and Sharma, D. (2020). Distributed information integration in convolutional neural networks. In *Proceedings of VISAPP*. INSTICC, SciTePress.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lenc, K. and Vedaldi, A. (2015). Understanding image representations by measuring their equivariance and equivalence. *CVPR*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Park, H. and Lee, K. M. (2016). Look wider to match image patches with convolutional neural networks. *IEEE Signal Processing Letters*, 24(12):1788–1792.

Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J. (2017). Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361.

Su, Y., Shan, S., Chen, X., and Gao, W. (2009). Hierarchical ensemble of global and local classifiers for face recognition. *IEEE Transactions on image processing*, 18(8):1885–1896.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Technical report, arXiv.

Xu, Y., Xiao, T., Zhang, J., Yang, K., and Zhang, Z. (2014). Scale-invariant convolutional neural networks. *CoRR*, abs/1411.6369.

Zhang, T., Zeng, Y., and Xu, B. (2016). Hcnn: A neural network model for combining local and global features towards human-like classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(01):1655004.

Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., and Ling, H. (2019). M2det: A single-shot object detector based on multi-level feature pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9259–9266.

Zheng, L., Yang, Y., and Tian, Q. (2017). Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244.