Multi-modal Information Extraction and Fusion with Convolutional Neural Networks

Dinesh Kumar
Faculty of Science and Technology
University of Canberra
Canberra, Australia
dinesh.kumar@canberra.edu.au

Dharmendra Sharma
Faculty of Science and Technology
University of Canberra
Canberra, Australia
dharmendra.sharma@canberra.edu.au

***Abstract*—Developing computational algorithms to model the biological vision system has challenged researchers in the computer vision field for several decades. As a result, state-of-the-art algorithms such as the Convolutional Neural Network (CNN) have emerged for image classification and recognition tasks with promising results. CNNs however remain view-specific, producing good results when the variation between test and train data is small. Making CNNs learn invariant features to effectively recognise objects that undergo appearance changes as a result of transformations such as scaling remains a technical challenge. Recent physiological studies of the visual system are suggesting new paradigms. Firstly, our visual system uses both local features and global features in its recognition function. Secondly, cells tuned to global features respond quickly to visual stimuli for recognising objects. Thirdly, information from modalities that handle local features, global features and color are integrated in the brain for performing recognition tasks. While CNNs rely on aggregation of local features for recognition, these theories provide the potential for using global features to solve transformation invariance problems in CNNs. In this paper we realise these paradigms into a computational model, named as global features improved CNN (GCNN), and test it on classification of scaled images. We experiment combining Histogram of Gradients (HOG) global features, CNN local features and color information and test our technique on benchmark data sets. Our results show GCNN outperforms traditional CNN on classification of scaled images indicating potential effectiveness of our model towards improving scale-invariance in CNN based networks.**

***Index Terms*—convolutional neural network, scale invariance, invariant features, global features, local features, histogram of gradients, color histogram**

I. Introduction

Whilst evolution has made our vision system a state-of-the-art biological object detector, recognition engine and classifier, making computer vision algorithms achieve the same remains a challenge and an active field of research. 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) [1]. CNNs have achieved great success in numerous computer vision tasks and are applied in various practical application domains such as in self driving cars, facial recognition authentication systems such as in mobile phones, medical image processing and quality assurance in manufacturing industries.

CNNs however still remain view-specific, producing good results when the variation between test and train data is small. This means when invariances are introduced in test images, CNN exhibit considerable drop in accuracy [2]–[4]. 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. The biological vision system of primates on the other hand learn invariant features and are hence able to recognise objects regardless of their pose, size or orientation. It remains a technical challenge to make CNNs learn invariant features to effectively recognise objects that undergo appearance changes as a result of transformations such as scaling.

Existing anatomical and physiological models of the visual system suggest visual stimuli captured on the retina propagates via low level cells to complex cells [5]–[8]. The dense low level cells such as V1 cells in the ventral stream are highly responsive to extracting low level local features such as lines, curves and their orientations, while complex cells such as the AIT aggregates these information into a higher representative form referred to as global features. Whilst this model suggests the final global features from AIT are delivered to the cortex for performing the vision task at hand, recent physiological studies are suggesting new paradigms on the workings of the visual system. Firstly, studies of the visual pathway reveal that the visual system uses both local and global features in its recognition function. Secondly, cells tuned to global features respond to visual stimuli much quickly than to cells tuned on local features leading to suggestions of a unique response strategy of the visual system to speed-up recognition. This suggests cells (neurons) tuned to global features can be activated independently by visual stimuli rather than waiting for information to propagate through low level cells first. And lastly, information from three modalities namely local features, global features and color information are integrated and collectively used for recognition purposes [9], [10]. Earlier works of Navon [11] and Avargues-Weber *et al.* [12] have suggested the use of global features as an important ingredient for visual recognition tasks.

CNNs are based on the local to global feature extraction strategy of the visual system and uses layers of convolution operations to extract features. While later layers in a CNN can be regarded as *complex cell layers* it is difficult to ascertain or segregate these layers from other layers. Hence for this

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

reason, we propose opening a new visual stimuli pipeline that explicitly extracts global features using global feature extraction methods. We would like to emphasise that in our work global features extracted at the end of a CNN feature extractor pipeline are different to the global features extracted by global feature extraction methods. Further, while it is still unclear how the visual system learns invariant features to solve invariance problems, we hypothesis that global features and color information play an important role in this function supported by evidence provided in Huang *et al.* [9]. While the idea of using global features for visual recognition task is not new, these theories do provide the potential for using global features to solve transformation invariance problems in CNNs. Also, we acknowledge there are several invariances to consider such as translations, rotations and color changes but for the purpose of this paper we consider solving scale invariance classification of images.

In this paper, we address improving classification of scaled images by exploiting these paradigms and propose an ensemble neural network model named as global features improved CNN - (GCNN). GCNN allows CNNs to better classify scaled images by combining both global and local features of the target image during network training. This is achieved by extracting global features from feature descriptor methods, local features from CNN independently and then fusing them with color information in the fully connected layer of the network. We experiment with Histogram of Gradients (HOG) [13] as our global feature descriptor method. For color information we use normalised color histograms.

We conduct extensive experiments to evaluate scale invariance performance of GCNNs. We use two well-known CNN architectures in our work as our benchmark models, namely VGG16 [14] and LeNet5 [1]. Tiny ImageNet [15] and Fashion-MNIST [16] are used as our benchmark datasets. First the datasets are trained on the CNNs to establish benchmark results for comparison. Then we transform the physiological model proposed by Huang *et al.* [9] into a 3-channel computational model referred to as GCNN. This is achieved by integrating CNN as local feature extractor in the first channel, HOG as global feature extractor in the second channel and a routine to extract normalised color histogram in the third channel. We train GNN on the same datasets using the same training constraints as the benchmark CNNs such as number of training epochs, learning rate, loss function and optimizer. We study the performance of GCNN in classifying image samples on specific scale categories. We generate these scale categories by selecting images from the test dataset and scaling them. In total we generate 7 scale categories. For consistency we use the same scaled samples on all models developed. In all our case studies, performance of GCNN are compared with the classification scores from benchmark VGG16 and LeNet5 models on all scale categories. Whilst, on the basis of our experimental results, VGG16 and LeNet5 models show some degree of recognising scaled images, in comparison GCNN demonstrates higher geometric invariance in terms of recognising more scaled images accurately despite changes in scale. Thus, in applications where handling objects with multiple scales is desired, GCNN will prove beneficial.

This paper aims to contribute to the body of knowledge towards finding effective solutions to classification of scaled images by:

1) showing the usefulness of combining global features and color information together with CNN features, using a computational model based on the 3-channel physiological model such as one proposed by Huang *et al.* [9],
2) showing a probable validation of the plausible model of the vision system that indicate information processed in three distinct modalities are used for recognition tasks, and
3) showing global features such as HOG as well as color information prove more useful when applied on 3-channel color images than on grey-scale images.

The rest of the paper is organised as follows: Section II reviews related work while Section III introduces our model. Section IV describes our experiment design and results are presented in Section V. We summarise and suggest future research directions in Section VI.

II. BACKGROUND

We summarise below the main advances in research in vision systems providing the basis for the current research and the presented outcomes.

A. Global Features as a Modality in the Vision System

Recent research by Park and Lee [17] show that humans tend to view wide areas around the target pixel to obtain spatial relationships between features. This claim is further supported by Huang *et al.* [9] through their research that the visual system in humans, non-human primates and honey bees are more sensitive to global features than local features. To test this hypothesis the authors trialled their experiments on behaving monkeys where they were trained to make a saccade to a target in the black background. These targets represented shapes using local features (solid shapes such as a circle) and global features (such as a hole in circle). Their experiments showed detecting a distinction or change in the global feature was faster than detecting a distinction or change in a local feature. This means the visual system uses information collected from wide areas around the target pixel (*global* features) to obtain spatial and semantic relationships for identifying objects prior to using local features. They also placed emphasis on the importance of color as a key modality in recognition of objects. Based on their findings they proposed a plausible model of the vision system as described in Fig. 1 from which we adopt layers (a), (b) and (c) since our main focus is recognition of scaled images.

B. Global Features

The term *global features* in computer vision refers to describing an image as a whole [18]. They are used to generalise the distribution of the visual information in the object through various statistics that represent information on contours, shape

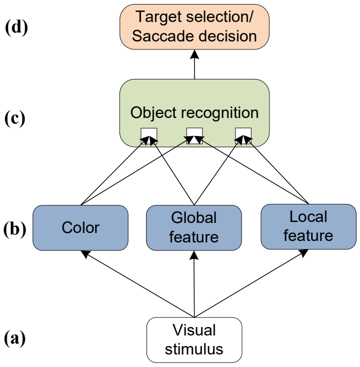

Fig. 1: A new model of vision proposed by Huang *et al.* [9]. It describes a 3-channel architecture processing color, global and local features as separate modalities (layer (b)), before combining the features from the modalities for object recognition (layer (c)).

and texture in the image and are useful for tasks such as object detection and classification. *Local features* are used to describe image patches (key points in the image) of an object [19]. These features represented as lines and curves are basic building blocks of object shapes and are useful for tasks such as object recognition.

Investigating the impact of incorporating global feature extraction strategy in CNNs has attracted some attention in the research community. For example, Zheng *et al.* [20] examined how pre-trained Alex-Net and VGG-19 networks process local and global features. Several studies further report use of global feature descriptors to solve image classification and object detection problems such as HOG [13], [21], invariant moments [22], [23], uniform local-binary patterns (LBP) and discrete cosine transform (DCT) [24], discrete fourier transform (DFT) [10], color and entropy statistics [19], and shape index [18] but mostly without combining with CNNs. A similarity between some of these work and our method is that features from global feature descriptors are extracted as separate modalities in parallel and later fused in the model architecture.

Another application of global features is demonstrated in the work of Kerdvibulvech and Saito [25]. However, in contrast to local-global feature extraction in CNNs, they proposed a sequential model where they applied global processing on the input image followed by local processing to extract features for detection of the positions of fingertips of a hand playing a guitar. In the context of their work, a global feature descriptor (Gabor filters) was applied twice - first on the entire hand skin area and then on a localised region of interest (ROI).

Our research reveals only few studies have shown combined use of global feature descriptors with CNNs. A combined invariant moments and CNN based approach for image analysis is proposed by Mahesh *et al.* [26]. However, instead of combining moments based global features with local features the authors use zernike moments to derive the initial training convolution kernel coefficients by changing the moment order. Their results showed zernike based kernels outperformed CNN architectures that used random kernels as initial training parameters on image analysis and classification.

Furthermore, a neural network model is proposed by Zhang *et al.* [27] called histogram of gradients improved CNN (HCNN) that combines texture features from traditional CNNs and structural features from HOG to cover the shortness of CNNs in recognising fooling images (where some local features are chaotically distributed). Fusion of global with local features made their network become more sensitive to fooling images. We follow a similar approach in our work and extend it to evaluate on scaled images.

C. Color

Apart from texture and shape information, color represents an important filter our vision system also uses for object recognition [9]. Making CNNs learn classification on color distribution is studied in the work of Rachmadi *et al.* [28]. The authors trained their CNN architecture on images converted to HSV and CIE Lab color spaces and successfully applied it for vehicle color recognition with improved performance over traditional CNNs. Their method however does not employ use of color as a separate modality representing global feature information that can be used with CNNs in parallel. In a similar work Chowda & Chen [29] explore color spaces and show that certain classes of images from their datasets are better represented in particular color spaces. In our work, we treat color as a separate modality for image information and instead of training images in different color spaces we extract and train normalised color histogram of images by fusing them with local and global features.

D. Scale-Invariant CNNs

In this space, studies have largely concentrated on creating pyramid based CNN architectures to handle scale-invariant classification such as image pyramids, feature pyramids and filter pyramids. In image pyramid based CNN, copies of the input image are generated at multiple scales forming a pyramid of images. They are processed in the convolution layer using the same filter, generating feature maps of different sizes (feature pyramid). These feature maps are then normalised to obtain the same spatial dimensions and pooled to obtain a locally scale-invariant representation. The work of Kanazawa *et al.* [30] and Xu *et al.* [31] follow this process. We note scaling images in this way is similar to applying scale augmentation to datasets. In our work we apply no augmentation.

By default, deep CNNs generate a conical hierarchy of feature maps (feature pyramid). However, in the simplest CNN there are no connections between feature maps of different layers. This property is studied in the work of Lin *et al.* [32] where they show that by developing lateral connections from each feature map in the pyramid makes them 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 [33]–[35].

Based on Google's INCEPTION model [36], several other similar models propose the use of kernels of different sizes

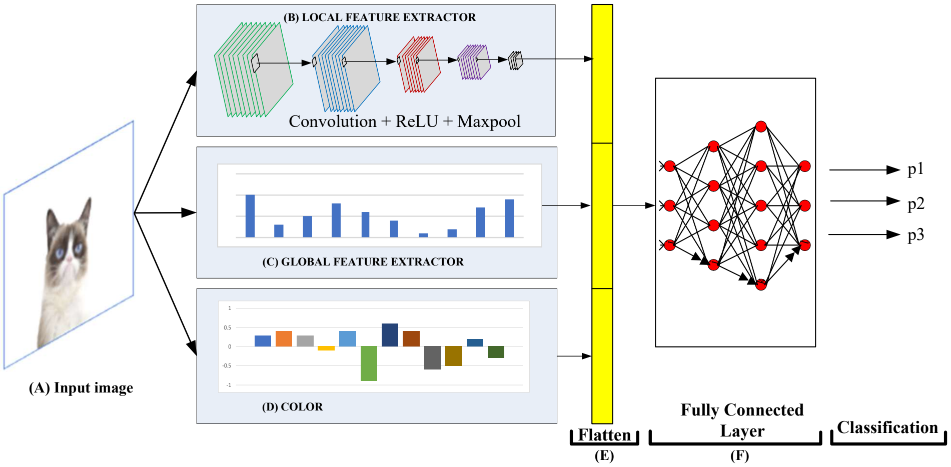

Fig. 2: Architecture of GCNN. First an input image (A) passes through a local feature extractor (B) producing a set of features maps. The same image is processed by a global feature descriptor function (C) producing a set of feature descriptors. A normalised color histogram is produced in (D). Outputs of (B-D) are fused as a single vector in the flatten layer (E) and forward propagated to the classifier (F).

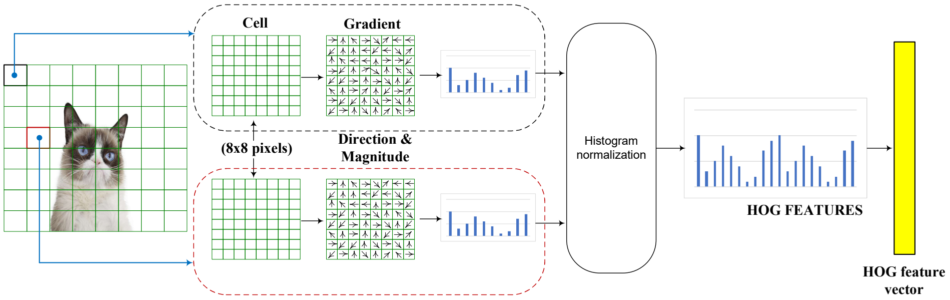

Fig. 3: Histogram of Gradients (HOG) descriptor flowchart.

forming a pyramid of filters [37]–[41]. The models use multi-scale filters to capture context from different spatial areas of the input features to improve image classification. Based on this concept, Kumar *et al.* [41] proposed a model in which they used two feature extraction channels dedicated to local and global processing of input data in parallel. They relied on the concept of using large kernels [42] in a multi-scale kernel setting in their global processing channel to extract features from relatively larger spatial area of the image. For local processing they used the LeNet5 [1] feature extraction layers. The outcome (features) from each channel is processed using a decentralised multisensory information integration model [43]. That is, the local and global features are first processed by a small neural network (local processor) separately and their outcomes processed by another central processor neural network for final classification. Their model showed promising results when tested on classification of scaled images. We follow a similar model architecture in our work but extend it to contain three channels to accommodate color information as a separate modality. In addition we remove the local processors and simply apply concatenation of features from the three channels for processing by a classifier.

Finally, Jaderberg *et al.* [2] introduced a trainable module called *Spatial Transformer* that uses a set of parameters that spatially transforms feature maps, and updates these parameters during the learning process, thus learning the spatial transformation that should be applied to the feature map. However, this technique limits the number of objects that can be modelled.

In our work, we avoid complicated pyramid based methods and adopt a standard CNN as our local feature extractor. Further, despite the existence of local, global and color feature extraction techniques, the fusion of outputs from these techniques using a physiological model of vision as such as the one proposed by Huang *et al.* [9] have not been tested for image classification. These approaches combined have also not been tested to ascertain the networks ability to be spatially invariant. This paper attempts to fill this gap.

III. MODEL

The ensemble GCNN model comprises of six main parts (A-F) as shown in Fig. 2. They are explained in the following sub-sections.

A. *GCNN*

Given the localised patch-wise convolution by small sized kernels (relative to the image size), CNNs lose the spatial and geometry information from images which may prove essential for learning invariance. To address this we focused on combining global features from training data. To this end we develop a novel model called global features improved CNN (GCNN) that comprises of three pipelines in its architecture. The pipelines are dedicated to extracting local, global features and color information from the input data independently (Fig. 2 (B-D)). In the local feature extractor pipeline (Fig. 2 B)) we deploy benchmark CNN. The function of the CNN is to form key point descriptions of the input images in the form of *local-global* features. We implement the HOG feature descriptor in the global feature extractor pipeline (Fig. 2 C)) in order to extract contour features from images. Color information from images is extracted in the form of normalised histogram (Fig. 2 D)). The latter two layers do not require any trainable parameters.

B. GCNN Forward Propagation Process

To achieve the forward pass function, an input image is processed in parallel in the local, global and color feature extractor pipelines respectively. The operation of the CNN part of the network remains standard filtering the image through several successive convolution, ReLU and max pooling layers and producing a set of final feature maps as output. The same batch of input images meanwhile are processed by the HOG descriptor generating a set of normalised gradient descriptors. A normalised color histogram is produced for the batch of input images. Output from all pipelines are combined and reshaped into a vector form in the flatten layer (Fig. 2 (E)). This vector forms the input to the fully connected neural network classifier in GCNN (Fig. 2 (F)).

C. GCNN Backward Propagation Process

The flatten layer plays a key role in the implementation of the backward function. It receives gradients from the network and *unstacks* or *slices* the gradients in the exact same dimensions and shape of the input feature maps it received during the forward pass from the local, global and color feature extractor pipelines. Since there are three pipelines in GCNN, the flatten layer returns three sets of gradients - CNN gradients, HOG gradients and color gradients. The gradients corresponding to the CNN pipeline (CNN gradients) are back propagated through the layers using chain rule derivative algorithm. Since there are no trainable parameters in the HOG and color pipelines they terminate in the flatten layer.

D. HOG Feature Descriptor

In an image, abrupt changes in colour intensities pack a lot of useful information such as on edges and corners which are used to describe shapes in computer vision. Using colour intensities in a localised portion of an image, HOG feature descriptor generates a distribution in the form of a histogram of directions of gradients (oriented-gradients) of every pixel as well as its magnitude. The HOG algorithm requires three important parameters. First, number of gradient orientations (B) which will represent the bins in the histogram. Each bin represents an angle from $0^\circ - 180^\circ$ in increments of $(180^\circ/B)^\circ$. Second, the number of pixels p which is used to divide an input image into small connected square regions called cells each containing $(p \times p)$ pixels. Also required is a parameter called block-size (b) that groups number of adjacent cells. Example, consider an image of size (32×32) pixels, $p = 4$, block-size $b = 4$ and $B = 8$. Using these numbers, for the given image there are 64 cells, $4 \times 4 = 16$ pixels in a cell and $4 \times 4 = 16$ blocks with 4 cells in a block. For each pixel in a cell HOG calculates its the gradient orientation and magnitude and using these updates the counts in the bin representing the angular orientation of the gradient. In our example this results in a 8-bin histogram for all cells. Given $b = 4$, the HOG algorithm combines 4 adjacent cells' histograms by normalising them. Finally the normalised group of histograms from all blocks represent the HOG feature vector H. In our example $H = 128$ which is $B = 8$ multiplied by the numbers of blocks (16). Fig. 3 shows the flowchart of the HOG descriptor method implemented in our work.

E. Color Histogram

A color histogram describes the distribution of colors for a whole image or for a region of interest within an image. The advantage of using a color histogram is that it is invariant to rotation, translation and scaling of an object [44]. The disadvantage is that it does not contain semantic and spatial information causing two images with different contents but similar histograms to be classified same. To overcome this, we use HOG descriptor in our work to extract spatial information. For a given image I, its color histogram C is defined as a vector $C = c[1], c[2], c[3], ..., c[i], ..., c[N]$ where i represents a color in the histogram, $c[i]$ is the number of pixels of color i, and N is the number of color bins in the histogram. In order to compare images of different scales, we normalise Vector C in the range $[-1, 1]$ given by $C' = (C - \mu\{I\})/\max\{I\}$ where $\mu\{I\}$ and $\max\{I\}$ are the mean and max value of all pixels in an image respectively.

IV. THE EXPERIMENTS

We describe the datasets, GCNN component architectures and selected parameters and our experimental design in the following sub-sections.

A. Dataset Description

We test GCNN on color images having red, green and blue (RGB) channels, on the basis that our vision system normally perceives a scene in color. However, we also wish to evaluate GCNN on grey-scale images and confirm whether results on color and non-color images are comparable. As such the following datasets were selected for the experiments.

a) Tiny ImageNet: A subset of the ImageNet dataset designed for visual object recognition research [45], Tiny ImageNet [15] consists of 100,000 training images, 10,000 validation images and 10,000 test images in color. The images are cropped and resized to 64×64 pixels in RGB color channels and are divided into 200 mutually exclusive classes,

TABLE I: Architecture of VGG16 [14] and LeNet5 [1] networks used in our experiments.

Model	Layers
VGG16	(conv 3x3x64) → (conv 3x3x64) → (maxpool 2x2) → (conv 3x3x128) → (conv 3x3x128) → (maxpool 2x2) → (conv 3x3x256) → (conv 3x3x256) → (conv 3x3x256) → (maxpool 2x2) → (conv 3x3x512) → (conv 3x3x512) → (conv 3x3x512) → (maxpool 2x2) → (conv 3x3x512) → (conv 3x3x512) → (conv 3x3x512) → (maxpool 2x2) → (fc 4096) → (fc 4096) → softmax
LeNet5	(conv 5x5x6) → (maxpool 2x2) → (conv 5x5x16) → (maxpool 2x2) → (conv 5x5x120) → (fc 84) → (fc 10) → softmax

with 50 images in each class in the validation set. There are no class labels provided for the test images, hence we use the validation set as test images.

b) Fashion-MNIST: The Fashion-MNIST (FMNIST) dataset [16] consists of 60,000 training images and 10,000 test images of fashion products from 10 categories. The sample images are grey-scale (1-channel) of size 28x28 pixels. The training and test batches have equal distribution of the number of samples from each class.

B. CNN Architectures and HOG Parameters

For benchmarking and local feature extractor part of GCNN we use VGG16 and LeNet5 CNN models as described below.

a) VGG16 Network: Proposed by Simonyan & Zisserman [14], VGG16 is a popular CNN model used by researchers in the computer vision field for image classification and segmentation tasks. It was originally trained on ImageNet dataset [45] that contains over 14 million images categorised into 1000 classes. It achieved top-5 test accuracy of 92.7% becoming the 1st runner-up in the ImageNet 2014 challenge classification task behind GoogLeNet. Several configurations of the VGG CNN exist, ranging from 11, 13, 16 and 19 weight layers. These configurations are labelled A-E and differ only in the depth. In our work we use configuration D that contains 16 weight layers comprising of 13 convolution and 3 hidden layers in the fully connected part of the network. All convolution layers are configured with 3 x 3 filter sizes. The network also uses maxpooling layers. Table I describes the architecture of the VGG16 network. We train this network on the Tiny ImageNet dataset. For each image, the final feature map size is $(512 \times 2 \times 2)$.

b) LeNet5 Network: Proposed by LeCun [1], the LeNet5 network in our work comprises of three sets of convolution layers and two max pooling layers. The architecture is described in Table I. We train LeNet5 on FMNIST dataset by setting the hyper-parameter *padding* for the first and second convolutional layers to 2 and 1 respectively. We use LeNet5 on FMNIST dataset due to the relatively small sizes of the images which are 28×28 pixels, and also research such as [46] show the feasibility of using LeNet5 on small sized image datasets. For each image, the final feature map size is $(120 \times 2 \times 2)$.

c) HOG Parameters: We setup our HOG descriptor in GCNN with 8 orientation bins, 4 pixels per cell and block-size of 1. On Tiny ImageNet dataset this resulted in 2048 gradient features per image and 392 features per image on FMNIST dataset. We implemented our HOG using *scikit-image* image processing library in Python.

d) Color Histogram Bins: Each pixel in our datasets is normalised to a value in the range $[0-255]$ causing 256 possible color values in each channel of the image. 256 therefore represents the number of bins for each channel in the color histogram. For the 3-channel Tiny ImageNet dataset, the normalised vector C' from each channel is combined as a single vector with length of 756 $(256*3)$. For the 1-channel FMNIST dataset the length of $C' = 256$.

TABLE II: Dimensions of the input components received by the flatten layer and size of the final output vector for each GCNN model.

Component	GCNN (Tiny ImageNet)		GCNN (FMNIST)	
	Feature size	1D vector conversion	Feature size	1D vector conversion
CNN features	512x2x2	2048	120x2x2	480
HOG features	2048	2048	392	392
Color features	256x3	768	256x1	256
Output vector		**4864**		**1128**

e) Flatten layer: Table II summarises the number of elements in the three input components feeding into the Flatten layer and the size of the final output vector for each GCNN trained on Tiny Imagnet and FMNIST datasets respectively.

C. Training Process

First we train the benchmark CNNs - VGG16 on Tiny ImageNet and LeNet5 on FMNIST datasets separately. This establishes our benchmark results against which we compare results of GCNN networks. Then in a similar fashion we establish results by combining the CNNs with HOG features and color features. Hence we obtain a total of four trained models for comparison (two models per dataset).

We perform end-to-end training of the VGG16 and VGG16 based GCNN networks on Tiny ImageNet dataset. We use partial transfer learning on the VGG16 network whereby pre-trained weights of the feature extractor part of the VGG16 network trained on the ImageNet [45] dataset are loaded in the model for initial training. The last layer in the classifier is replaced with a new layer containing 200 neurons to match the number of classes in the Tiny ImageNet dataset. The weights of the classifier layers are reinitialised. During training all model weights are updated. The model is trained for 50 epochs with a fixed learning rate of 10^{-4}. For training LeNet5 and LeNet5 based GCNN on FMNIST dataset, the learning was adjusted to 10^{-1} for 2 epochs, 10^{-2} from epochs 3-50 and decreasing it to 10^{-3} for the rest of training. These models were trained from scratch for 100 epochs.

On all models, stochastic gradient decent and cross-entropy are used as learning and loss functions respectively, weight decay of 10^{-4} and momentum of 0.9. For training we use batch size of 4 and 1 for testing. We implemented our models

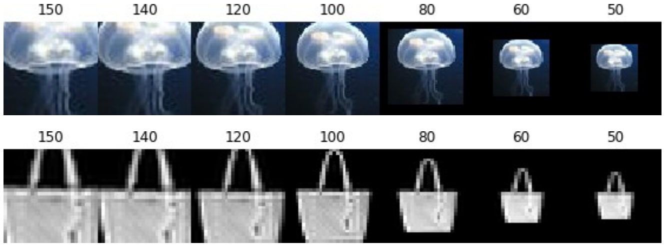

Fig. 4: A sample scaled test image from datasets Tiny ImageNet - n01910747 (jellyfish) (*top*) and FMNIST - bag (*bottom*). The numbers above each image indicate the scale factor (in %). Scale factor of 100 indicates no scaling.

TABLE III: Train losses and test accuracy on VGG16 based models trained on Tiny ImageNet dataset.

Model	train loss	test acc	difference
VGG16	0.018	0.576	
GCNN (VGG16+HOG+Color)	**0.012**	**0.586**	**-0.006 (loss)** **+1.0% (acc)**

TABLE IV: Train losses and test accuracy on LeNet5 based models trained on FMNIST dataset.

Model	train loss	test acc	difference
LeNet5	1.535	0.899	
GCNN (LeNet5+HOG+Color)	**1.491**	**0.903**	**-0.044 (loss)** **+0.4% (acc)**

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.

D. Preparing Scaled Images for Testing GCNN

We prepared scaled images for testing following the approach proposed in [41]. First, 7 scale categories are established - $[150, 140, 120, 100, 80, 60, 50]$. Each number indicates the scale factor (in percentage) applied to a test image. Numbers > 100 indicate enlargement while < 100 indicate reduction in the size of an image. We select at random 20 and 100 images per class from Tiny ImageNet and FMNIST test datasets respectively. Each image is then scaled according to the sizes defined in the scale category list, resulting in an additional 6 scaled images per class in addition to the original image (of scale 100%). In this way, for each class, 7 scale category folders are created and respective scaled images stored in them accordingly. Using this process, 140 scaled images per class are sampled from Tiny ImageNet dataset and 700 images per class from the FMNIST dataset. Following the process defined in [41], we create an *ensemble* dataset by combining all scaled images from all classes per dataset. This resulted in a total of $28,000$ scaled images for testing on Tiny ImageNet ($200 classes \times 140$), and 7000 scaled images on FMNIST dataset ($10 classes \times 700$). We analyse our models on scaled images from each of these scale categories independently as well as on the ensemble dataset (Section V-B). Fig. 4 shows an example image from each dataset and its corresponding scaled versions for testing.

E. Evaluation Metrics

We use *accuracy* as a performance measure to evaluate the generalisation capability GCNN and the benchmark models by finding out the total number of scaled images that were correctly classified in the respective scale categories.

V. Results and Discussion

A. Comparing Train and Test Statistics on Regular Images

Tables III and IV compare the train losses and test accuracy for all the networks on Tiny ImageNet and FMNIST datasets respectively. These statistics are the result of training the models using the training parameters outlined in Section IV-C and on raw images without any form of scale transformations. Our ensemble GCNN model outperforms the traditional benchmark CNN networks on all train and test metrics (indicated in bold). The highest test accuracy increase of 1.0% is recorded on GCNN combining VGG16, HOG and color features on Tiny ImageNet dataset. A similar trend is evident on the performance of GCNN on grey-scale FMNIST dataset using LeNet5 model.

From these results we derive two conclusions:

1) Combining local and global feature information in network training is useful in improving the classification accuracy of the models.
2) Applying HOG and color histogram on images that have 3-color channels and using that information in network training show better results when compared to the same on grey-scale images that have 1-color channel.

B. Robustness of GCNN on Different Scale Categories

Tables V and VI outline the classification results of our models on different scale categories and on different datasets. The statistics are obtained by testing the scaled images from each scale category on each model as well as all the combined images in the ensemble dataset. Similar to the analysis of [41], we count the number of scale categories (excluding the results of the ensemble dataset) where GCNN statistics are higher than the benchmark results. We refer to this count as *hit-rate*. For GCNN to show promise in classification of scaled images we set a minimum threshold of 50%, that is GCNN should at least perform better on 50% of the scale categories compared to the benchmark models.

On the basis of classification accuracy results of the models trialled in our work over various scale categories, we are able to demonstrate a superior performance of GCNN compared to the respective benchmark VGG16 and LeNet5 networks on both datasets. GCNN performed better on all scale categories on Tiny ImageNet dataset, where hit rate is equal to 100%, meaning the model was able to identify a high number of samples in its correct class in each scale category despite the images being scale transformed. We also observe that on Tiny ImageNet dataset, GCNN performance is better on both enlarged and reduced scaled images. Upon comparing classification accuracies of GCNN over LeNet5 on FMNIST

TABLE V: Performance summarization of VGG16 and GCNN networks on all the scale categories on Tiny ImageNet dataset.

		scale categories								
Model	**metric**	**ensemble**	**150**	**140**	**120**	**100**	**80**	**60**	**50**	**hit rate**
VGG16	acc	0.351	0.398	0.450	0.534	0.580	0.307	0.114	0.055	
GCNN (VGG16+HOG+Color)		**0.367**	**0.425**	**0.479**	**0.548**	**0.594**	**0.320**	**0.139**	**0.069**	1.000 (7/7)

TABLE VI: Performance summarization of LeNet5 and GCNN networks 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	
GCNN (LeNet5+HOG+Color)		**0.632**	0.570	0.640	0.772	**0.910**	**0.726**	**0.441**	**0.362**	0.571 (4/7)

dataset, we note the model only performed better on reduced scaled images. A probable explanation we present is that color in an image provides more contour and edge information that are useful for learning. Therefore the only useful information to learn from grey-scale images are the edges of objects (boundary between the light and dark pixels). The presence of the entire object in the reduced grey-scale images allowed the model to classify them with higher accuracy. On the other hand, the scaled-up grey-scale images lost parts of the object due to truncation and hence losing edge information causing the model to fail classifying them correctly. In contrast, in color images, despite truncation of parts of the object due to scaling up, enough detail still remains due to color variation for possible accurate classification. Here we also, relate that HOG extracts gradients from pixel intensities. RGB channels in color images allows HOG to extract more discriminatory features than 1-channel grey-scale images hence contributing to better results. Similarly, RGB channels in color images contributes to more information in the color histogram than grey-scale images. Further, GCNN outperformed the benchmark models on the ensemble test dataset, with a difference of 1.6% and 2.1% on Tiny ImageNet and FMNIST datasets respectively. This equates to 448 and 147 more scaled images classified correctly from the Tiny ImageNet and FMNIST ensemble test datasets respectively.

From the above analysis, we arrive at three conclusions:

1) The generalisation capability of CNNs can be improved by combining global features and color information with CNN local features. The same can be inferred on classification of scaled images whereby combined information from all 3 modalities as suggested by Huang *et al.* [9] prove beneficial.
2) Global features such as HOG as well as color information prove more useful when applied on 3-channel color images than on grey-scale images.
3) The statistics show models perform maximum best when processing images with no scale transformation applied. The accuracy progressively drops as higher degree of scaling is applied from the base scale category of 100%. This indicates the models are only view-specific and are highly tuned on the training dataset images. Although the proposed GCNN model performs better than the benchmark CNNs, the models still lack learning invariant features.

VI. Conclusion

In this work we develop a computational model based on a plausible model of the vision system proposed by Huang *et al.* [9]. Their model indicates the vision system uses global and color features alongside local features for object recognition tasks. Based on this architecture we propose a model called GCNN. We compared the performance of GCNN with benchmark CNN models and benchmark datasets. Also in this work we investigated whether global and color feature information can be used during network training to make CNNs handle spatial invariance problems better. As such we developed a case study of evaluating GCNN on scaled images. From our experimental results we conclude the generalisation capability of CNNs improve by fusing spatial information from images in the form of global and color information. There are also improvements observed on the networks ability in handling scaled images.

Problems and opportunities identified from the current project that require further investigation include a) to test other global feature descriptors with CNNs such as image moments, b) experiment with other color spaces in combination with RGB color space, c) test this technique to evaluate other forms of transformations such as rotations and translations and d) to generalise the proposed approach for other data sets.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025.

[3] E. Kauderer-Abrams, "Quantifying translation-invariance in convolutional neural networks," *arXiv preprint arXiv:1801.01450*, 2017.

[4] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," *CVPR*, 2015.

[5] T. Serre, "Hierarchical Models of the Visual System," in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds. New York, NY: Springer New York, 2013, pp. 1–12.

[6] T. Poggio and T. Serre, "Models of visual cortex," *Scholarpedia*, vol. 8, no. 4, p. 3516, 2013, revision #149958.

[7] P. M. Bays, "A signature of neural coding at human perceptual limits," *Journal of Vision*, vol. 16, no. 11, pp. 4–4, 09 2016. [Online]. Available: https://doi.org/10.1167/16.11.4

[8] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *J. Physiol*, vol. 148, pp. 574–591, Apr. 1959.

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

[10] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Transactions on image processing*, vol. 18, no. 8, pp. 1885–1896, 2009.

[11] D. Navon, "Forest before trees: The precedence of global features in visual perception," *Cognitive psychology*, vol. 9, no. 3, pp. 353–383, 1977.

[12] A. Avargues-Weber, A. G. Dyer, N. Ferrah, and M. Giurfa, "The forest or the trees: preference for global over local image processing is reversed by prior experience in honeybees," *Proceedings of the Royal Society B: Biological Sciences*, vol. 282, no. 1799, p. 20142384, 2015.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 886–893.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] F. F. Li, A. Karpathy, and J. Johnson, "Tiny ImageNet Visual Recognition Challenge," https://tiny-imagenet.herokuapp.com/, 2020, [Online; accessed 30-Dec-2019].

[16] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv, Tech. Rep., 2017.

[17] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1788–1792, 2016.

[18] D. A. Lisin, M. A. Mattar, M. B. Blaschko, E. G. Learned-Miller, and M. C. Benfield, "Combining local and global image features for object class recognition," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*. IEEE, 2005, pp. 47–47.

[19] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Combining global and local features for food identification in dietary assessment," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 1789–1792.

[20] Y. Zheng, J. Huang, T. Chen, Y. Ou, and W. Zhou, "Processing global and local features in convolutional neural network (cnn) and primate visual systems," in *Mobile Multimedia/Image Processing, Security, and Applications 2018*, vol. 10668. International Society for Optics and Photonics, 2018, p. 1066809.

[21] T.-K. Nguyen, M. Coustaty, and J.-L. Guillaume, "A combination of histogram of oriented gradients and color features to cooperate with louvain method based image segmentation," in *VISIGRAPP 2019*, 2019.

[22] J. Wu, S. Qiu, Y. Kong, Y. Chen, L. Senhadji, and H. Shu, "Momentsnet: a simple learning-free method for binary image recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2667–2671.

[23] S. Zekovich and M. Tuba, "Hu moments based handwritten digits recognition algorithm," in *Recent Advances in Knowledge Engineering and Systems Science*, 2013.

[24] S. Margae, M. Ait Kerroum, and Y. FAKHRI, "Fusion of local and global feature extraction based on uniform lbp and dct for traffic sign recognition," *International Review on Computers and Software (IRECOS)*, vol. 10, 01 2015.

[25] C. Kerdvibulvech and H. Saito, "Vision-based detection of guitar players' fingertips without markers," in *Computer Graphics, Imaging and Visualisation*, ser. Computer Graphics, Imaging and Visualisation: New Advances, CGIV 2007, dec 2007, pp. 419–424, computer Graphics, Imaging and Visualisation: New Advances, CGIV 2007 ; Conference date: 13-08-2007 Through 16-08-2007.

[26] V. G. Mahesh, A. N. J. Raj, and Z. Fan, "Invariant moments based convolutional neural networks for image analysis," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 936–950, 2017.

[27] T. Zhang, Y. Zeng, and B. Xu, "Hcnn: A neural network model for combining local and global features towards human-like classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 01, p. 1655004, 2016.

[28] R. F. Rachmadi and I. Purnama, "Vehicle color recognition using convolutional neural network," *arXiv preprint arXiv:1510.07391*, 2015.

[29] S. N. Gowda and C. Yuan, "Colornet: Investigating the importance of color spaces for image classification," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 581–596.

[30] A. Kanazawa, A. Sharma, and D. W. Jacobs, "Locally scale-invariant convolutional neural networks," *CoRR*, vol. abs/1412.5104, 2014. [Online]. Available: http://arxiv.org/abs/1412.5104

[31] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang, "Scale-invariant convolutional neural networks," *CoRR*, vol. abs/1411.6369, 2014. [Online]. Available: http://arxiv.org/abs/1411.6369

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

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

[34] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9259–9266.

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

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[39] Z. Liao and G. Carneiro, "Competitive multi-scale convolution," *arXiv preprint arXiv:1511.05635*, 2015.

[40] H. Wang, A. Kembhavi, A. Farhadi, A. L. Yuille, and M. Rastegari, "Elastic: Improving cnns with dynamic scaling policies," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2258–2267.

[41] D. Kumar and D. Sharma, "Distributed information integration in convolutional neural networks," in *Proceedings of VISAPP*, INSTICC. SciTePress, 2020.

[42] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[43] W.-H. Zhang, A. Chen, M. J. Rasch, and S. Wu, "Decentralized multisensory information integration in neural systems," *Journal of Neuroscience*, vol. 36, no. 2, pp. 532–547, 2016.

[44] R. S. Choras, "Image feature extraction techniques and their applications for cbir and biometrics systems," *International journal of biology and biomedical engineering*, vol. 1, no. 1, pp. 6–16, 2007.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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