New Color Fusion Deep Learning Model for Large-Scale Action Recognition

Yukhe Lavinia

Department of Computer Science
California State University
Fullerton, CA 92831, USA
ylavinia@csu.fullerton.edu

Holly Vo

Department of Computer Science
California State University
Fullerton, CA 92831, USA
hhvo@csu.fullerton.edu

Abhishek Verma

Department of Computer Science
New Jersey City University
Jersey City, NJ 07305, USA
averma@njcu.edu

Abstract: In this work we propose a fusion methodology that takes advantage of multiple deep convolutional neural network (CNN) models and two color spaces RGB and oRGB to improve action recognition performance on still images. We trained our deep CNNs on both the RGB and oRGB color spaces, extracted and fused all the features, and forwarded them to an SVM for classification. We evaluated our proposed fusion models on the Stanford 40 Action dataset and the People Playing Musical Instruments (PPMI) dataset using two metrics: overall accuracy and mean average precision (mAP). Our results prove to outperform the current state-of-the-arts with 84.24% accuracy and 83.25%mAP on Stanford 40 and 65.94% accuracy and 65.85% mAP on PPMI. Furthermore, we also evaluated the individual class performance on both datasets. The mAP for top 20 individual classes on Stanford 40 lies between 97% and 87%, on PPMI the individual mAP class performance lies between 87% and 34%.

Keywords: Deep Convolutional Neural Networks; Deep Learning Fusion Model; Action Recognition; VGGNet; GoogLeNet; ResNet.

Biographical notes: Yukhe Lavinia received her MS degree in Computer Science from California State University, Fullerton. Her research interests are in deep learning, computer vision, model ensembling, and image recognition.

Holly Vo received her MS degree in Computer Science from California State University, Fullerton. Her research interests are in deep learning, computer vision, model ensembling, and image recognition.

Abhishek Verma received his Ph.D. in Computer Science from New Jersey Institute of Technology, NJ, USA. He is presently Associate Professor of Computer Science at New Jersey City University, NJ, USA. His research interests are within the broad area of data science, big data, and machine learning. Deep learning on big datasets such as deep convolutional nets, model ensembling, image/video/speech recognition, natural language processing, financial market analysis, sentiment analysis. Computer vision on big datasets in video and image. Fusing multiple modalities from video/images/text/speech. Data mining, artificial intelligence, and biometrics for surveillance and security.

1 Introduction

With the rise of search engines technology and its pervasive use to obtain information about literally everything, action recognition, which With the rise of search engines technology and its pervasive use to obtain information about literally everything, action recognition, which finds direct application in information retrieval, becomes more relevant than ever. Over the recent years, deep learning in computer vision has gained significant advances in solving many image classification problems. These successes have led to applications in various domains: traffic and vehicle surveillance (2), biomedical image classification (3), remote sensing (1), robot navigation (5), and food product quality inspection (4).

As image classification, object detection, and action recognition are closely intertwined problems, advances in any one of these help to improve the other two areas. The annual Imagenet Large Scale Visual Recognition Challenge (ILSVRC) (7) is one of the most popular arenas on which researchers present their new and innovative methods. With the success of AlexNet (8) in 2012, deep convolutional neural networks (CNN) have enjoyed rising popularity as more researchers dedicated their works on it. The best performing groups on ILSVRC in 2013, 2014, and 2015 used a type of deep learning as their primary methods. Following this trend, we draw inspiration from these top performance deep learning models, namely GoogLeNet (13), VGGNet (16), and Residual Nets (17). We build our proposed models by fusing these three deep learning models.

We aim to improve action recognition performance in still images. We show the following: (1) That fusion of the multiple deep learning models improves classification performance as these collective vectors show superior discriminatory power over individual model, (2) That training these models on the oRGB color space dataset and adding their extracted features further improve the performance.

The following describes the organization of this paper: Section II discusses previous works on action recognition, section III presents a brief review on GoogLeNet, VGGNet, and ResNet, section IV describes the datasets we used in our experiments, section V presents the details of our methodology, section VI includes our experiments and results, and finally section VII concludes the paper and discusses future work.

2 Related works

The publicly available action recognition datasets are either in the form of videos or still images. Video-based action recognition have received more attention lately, although studies dedicated to its still image-based counterpart is not too far behind. The most popular action

recognition method uses object detection and image segmentation. This method commonly makes use of the annotations provided on the datasets to generate human bounding boxes that facilitate action classification.

Fusion of multiple features have been studied as a consistent way to improve action recognition performance (24), (25), (37). The idea is that by aggregating more features we can collect more information that we can use to better characterize the action. Additionally, it is expected that multiple features would complement each other, which then would improve upon the discriminatory power of the individual model.

Simonyan and Zissserman (18) proposed two-stream spatiotemporal networks that take advantage of the temporal information contained in the sequence of video frames while also processing the spatial information as regular CNNs do. The model takes video frames as input and decompose them into their space and time components. Note that “space" or “spatial" refers to a single video frame that is effectively treated as a still image, and “time" or “temporal" component refers to the motion captured in multiple video frames, thus preserving the continuity between the frames. The two components become separate networks (streams) that perform video action recognition. While the spatial stream operates like a usual CNN, the temporal stream employs optical flow stacks to determine motion between video frame sequences. Both streams' CNN architecture take the form of VGG-CNN-M-2048 (5 convolutionals, 3 fully-connnected layers) as proposed in (19). To apply fusion, Simonyan and Zisserman conducted separate experiments: 1) by taking the average of both networks' softmax score results, and 2) by extracting the two networks' softmax layer features and applying multi-class linear SVM on these vectors.

Rosenfeld and Ullman (20) employs the fusion method to extract features from the following: 1) the global average pooling layer of VGG-GAP (30), 2) the FC6 layer of VGG-16 (16), and 3) the pool5 layer of ResNet-151 (17). The extracted features are used to accomplish the first of the two steps in their action classification process. Their overall method consists of: 1) concept recognition, and 2) action recognition. The first step uses an additional dataset called “concept" dataset that contains numerous images labeled with common concepts or keywords. This concept dataset is used primarily to train a classifier purposed to recognize common concepts (thus called concept classifier). Using the images' extracted features, the concept classifier is used to generate concept score vectors. The second step is the actual action classification and uses the action dataset. The concept score vectors are used to train another classifier that performs action classification on the targeted action dataset.

It is also common to use a combination of image segmentation and action mask to improve accuracy. Zhang *et al*. (26) proposed the use of action mask to isolate the human-object interaction from the background, claiming that accurate action recognition can be done without the help of human bounding box. Based on the observation that interaction between object parts hold significant clues regarding the action in the image, the action mask is obtained by 1) generating objects using selective search (27), 2) extracting the object parts, and 3) discovering the dependency and association of these parts using the image pixel RGB values, which is based on the assumption that spatially close objects are related to each other. The resulting action mask is a minute shape of the human-object interaction. Action presentation is constructed by taking the action masks to weed out object proposals generated in the very first step, concatenating all the object parts into high-dimensional Fisher vectors, and performing product quantization (28). A one-vs-all linear SVM is learned to perform action classification.

Zhao *et al.* (38) proposed a "generalized symmetric parts model" (GSPM) that improves upon the traditional method bag-of-words (BoW) (9) to locate semantically meaningful regions, primarily because the features in this region are more discriminative. The overall framework includes several steps: 1) edge detection, 2) region segmentation, 3) mask generation, 4) feature extraction, and 5) BoW model creation. GSPM, which is developed independently, is used on step 3 to generate action masks. GSPM is based on two statistically supported observations: 1) that hands and feet (or arms and legs) are two pairs of body parts that contribute significant clues regarding an action; hence the “symmetric" aspect of GSPM mainly refer to these pairs, and 2) that semantically meaningful regions enclose these symmetric parts, as suggested by the close spatial relationship between them. The symmetric pairs' location are used as foci for ellipses to model the interaction between them. Zhao *et al.* added one more element, “spine," which refers to the distance between the symmetric pairs. These three elements are used to distinguish actions as the ellipse size and the spine length would vary for each action. Zhao *et al.* employs four types of feature representations: the 7 scale SIFT (10), color name (CN) (37), early fusion between the shape resulting from SIFT and color from CN, and late fusion between the shape and color, which is done after vocabulary assignment. Steps after feature extraction follow the traditional BoW method.

3 Overview of our base deep learning models

Our fusion method uses deep CNNs as base. We chose the winners of the ILSVRC 2014 and 2015: GoogLeNet, VGGNet, and ResNet. The following presents an overview of each deep CNNs and also the CNN model version that we ended up using for our fusion.

3.1 GoogLeNet

The winner of ILSVRC 2014, "GoogLeNet" (13), is developed based on two main ideas: 1) sparsity (14), and 2) dimensionality reduction. The sparsity principle is based on a study showing that given a sparsely structured network, we can optimally build a network topology by analyzing the output and correlation of clustered neurons. We create the succeeding layer's units by grouping highly-correlated neurons. Computation is performed by the dense matrices covering the newly-created sparse units.

The second principle, dimensionality reduction, is one of the most common techniques in machine learning. Its application in Inception-v1 is inspired by Network-in-Network (15), which introduces 1x1 convolutional kernels that perform both dimensionality reduction and feature maps computation across multiple channels.

The Inception-v1 block consists of 1x1, 3x3, 5x5 convolutions and a 3x3 max pooling layer that are arranged in parallel. The 1x1 kernel precedes the 3x3 and 5x5 convolutional layers and succeeds the 3x3 max pooling layer. Each convolution includes the rectified linear unit activation.

GoogLeNet also employs two auxiliary classifiers in the form of a small convolutional networks with a dropout ratio. During training the auxiliary classifiers generate two loss functions that are concatenated with the final loss function at the end of the overall network.

The overall GoogLeNet architecture that we used in our experiments consists of multiple Inception modules with some max pooling layer inserted between the Inception modules, an average pooling layer (as suggested in (15)) to avoid overfitting, a fully connected layer, a dropout layer, and a softmax layer.

3.2 VGGNet

The VGGNet (16) achieved second place at the ILSVRC 2014. It is a uniform deep network (16 and 19 layers) that employs 3x3 receptive fields throughout the whole network. The use of 3x3 kernel serves two purposes: 1) to increase the discriminative power of the rectified linear activation since using 3x3 requires having more layers (two or three layers as opposed to one, as in the case of 7x7 kernel), and 2) to decrease the number of parameters.

In our work, we chose to use the 19-layer version, although we did a preliminary benchmarking using the 16-layer version. The 19-layer model consists of two blocks of two-layer 3x3 convolutions, three blocks of four-layer 3x3 convolutions, several max pooling layers before and after these convolution blocks, three fully connected layers, and a softmax layer.

3.3 Residual Network (ResNet)

The current state-of-the-art and winner of ILSVRC 2015, the Residual Networks or ResNets, is one of the deepest neural networks, if not the deepest with its current incarnation of 1001 layers (33). ResNet (17) employs skip connections that perform identity mapping to achieve the desired output. For each residual unit, an input x is connected to a weight layer. This connection is branched: one branch undergoes a residual function F(x) and then the activation function, while another bypasses all these weight layers, hence the name âŁœshortcut" or âŁœskip" connection. Since x retains its value by skipping the function, it is called âŁœidentity." The purpose of this identity shortcut path is identity mapping, allowing x to add to F(x), thus forming F(x) + x. He et al. argued that this method offers an easier way to optimize the network, mainly because we only need to add the learned output of the stacked layer (the "residual"), to the identity x to get the desired output.

The residual network implementation inserts batch normalization (BN) (32) between each convolution and rectified linear unit. The purpose is to reduce the internal covariate shift, that is, the changes on input distribution in layers resulting from changes on activation values during training. In deep neural networks such as residual network, a small change in activations and input distribution in early epochs could lead to great changes in the later epochs, which in turn would impact accuracy. BN provides a solution by normalizing each mini-batch.

Based on depth, ResNet comes in many variations. We used the 50, 101, and 152 layers since we wanted to take advantage of the increased depths as we expected them to yield better results.

4 Description of Datasets

4.1 Stanford 40 Action Dataset

We used the Stanford 40 Actions dataset (6) to evaluate our models. The dataset has 40 action classes on activities such as washing dishes, playing a violin, playing a guitar, cooking, running, riding a horse, etc. Stanford 40 Action consists of 9,532 images, with 4,000 images for the train set and 5,532 for the test set. Since it does not come with a validation set, we took 800 images from the training set to make up a validation set. Thus, we have 3,200 images for training, 800 images for validation, and 5,532 images for testing.

Figure 1 Example images of Stanford 40 Actions dataset (6).

Our train/validation sets have uniform number of images per class, with the train set having exactly 80 images per class and our validation having 20 images per class. The train/validation images vary in resolutions (the smallest is 200x200 and largest 989x600) and sizes, ranging from about 4KB to 170 KB. Unlike the train/validation sets, our test set has different image count per class, ranging from 82 to 196 images. The test images also have various resolutions (the smallest is 200x200 while largest is 997x600) and sizes (smallest is 5KB and largest 156KB).

Stanford 40 is known as one of the most challenging still image action recognition datasets, mainly due to its large number of categories, various occlusions, angles, visibility, poses, and background clutter. Fig. 1 depicts example images of the Stanford 40 Actions. We see a large variety of background clutters and human body part visibility as many images show only certain body parts and not the whole body. Another thing to note is that certain small interactive objects such as phone or cigarettes are often wrapped in a person's fingers, which adds to the level of occlusion. However, there are certain interactive objects that are almost always fully visible, or at least sufficiently visible to be recognized due to its bigger size. Examples of these include horse, fishing rod, and bike. As we see later, this element would have impact on individual class accuracy.

4.2 PPMI dataset

The second dataset we used is the People-playing-musical-instruments (PPMI) dataset (34). The images capture humans and their interaction with twelve different musical instruments: bassoon, cello, clarinet, erhu, French horn, harp, recorder, flute, guitar, violin, trumpet, and saxophone. The 12 types of instruments are being portrayed twice: in one image it is being played by a human (labeled PPMI+), and in another it is merely being held by a human and not played (PPMI-). There could also be multiple different instruments and multiple people in the images. For our experiments, we treat each instrument's PPMI+ and PPMI- images as different classes. Thus, since we have 12 types of instruments and each type has a PPMI+ and PPMI- images, we have 24 classes.

The PPMI dataset consists of 4,209 total images. We used 1,686 images for our train set, 424 images for validation, and 2,099 images for test. The number of image per class varies in all train, validation, and test sets. We used between 55 to 80 images per class for training, 14 to 20 images per class for validation, and 64 to 100 images per class for testing. Like Stanford 40, PPMI's image resolution and size come in a great variety as well. The smallest resolution is 66x100 with 3KB while the largest 4288x2848 with 1.1MB.

5 Proposed Deep Convolutional Neural Network Fusion Methodology

In this work, we propose a methodology that is built upon the late fusion of features extracted from different deep convolution neural networks (CNNs) on multiple color spaces. With this methodology, we examine the power of heterogeneous fusion of deep neural networks and color spaces. In particular, GoogLeNet, VGG-19, ResNet-50, ResNet-101, and ResNet-152 are separately benchmarked for two aforementioned action datasets with pre-trained weights that each net has learned on the ImageNet dataset.

5.1 Color Spaces and Color Fusion

Each deep CNN will be trained on two color spaces: the common color space RGB and the opponent color space oRGB. The oRGB color space has proven its discriminative power via texture features on KTH-TIPS2 material datasets and MIT scene dataset in the work of Banerji *et al.* (35). As each color space carries different characteristics and prevails on selective visual task, this work will evaluate the role of oRGB and the power of its fusion with RGB in the fusion of learned features from convolution neural networks.

While the fundamental RGB color space is composed of three color components that are closed to red, green, and blue wavelengths, oRGB is built on three psychological opponent axes: white-black, red-green, and yellow-blue. The original oRGB is derived from RGB in two steps. The first step linearly transforms RGB to a color space of 3 axes: white-black, yellow-blue, and magenta-red/cyan-green. The second step does a rotation around the first axis to finally create a true opponent color spaces (36). Similar to (35), the second step is omitted in this work for computational simplicity. Thus, three color channels L, C_1, and C_2 of oRGB are transformed from RGB as follows:

$$\begin{aligned} L &= 0.2990R + 0.5870G + 0.1140B \\ C_1 &= 0.5000R + 0.5000G - 1.0000B \\ C_2 &= 0.8660R - 0.8660G \end{aligned} \tag{1}$$

5.2 Image Preprocessing

As raw images of both datasets are captured in varying resolutions, standardizing image size is required before they are fed into any CNN. Whole color images on either RGB or oRGB color space are uniformly squashed into a fixed 256x256 size. Although both datasets are provided with annotated bounding boxes around actions, we decide to experiment on the whole image to preserve action background. The decision was made after comparisons of individual CNN performance on whole images against bounding boxes, with the CNNs trained on whole image outperforming those trained on bounding boxes. Each CNN implicitly crops squashed 256x256 images to 224x224 before passing them through its convoluted layers. Fig. 2 shows sample pre-processed images from PPMI dataset on both RGB and oRGB color spaces.

5.3 CNN Feature Extraction

For each net that was trained on a specific color space for a dataset of n actions, we choose to extract features from either the last fully connected (FC) layer or the softmax layer for two reasons. First, these layers consistently output n-dimension vector for each image across

Figure 2 Example pre-processed images of PPMI dataset (34) on RGB and oRGB color spaces.

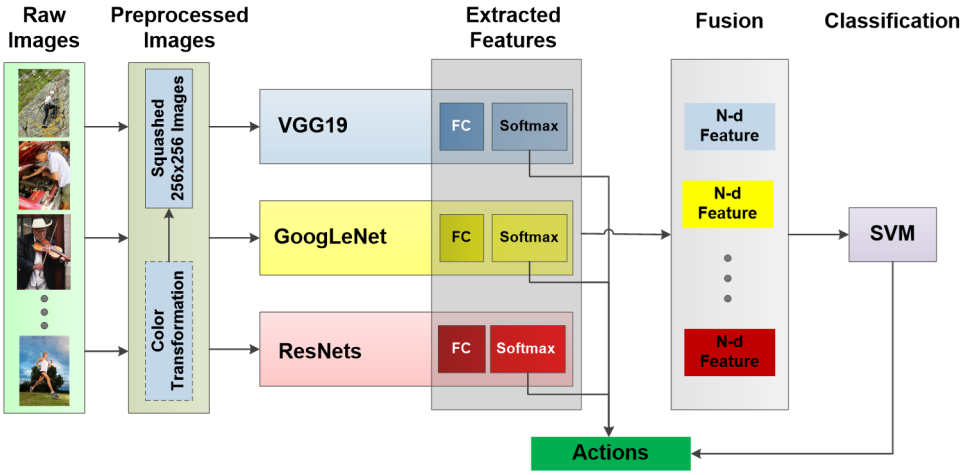

Figure 3 Overview of proposed color fusion deep learning methodology.

CNNs. Secondly, our preliminary experiments showed that features extracted from these late layers significantly yield better results than those from preceding layers.

The final feature is created by concatenating extracted n-D features of different CNNs on one or both color spaces. As extracted feature size is identical for all nets, the roles of individual CNNs in each feature fusion are all equal. Performance of each fusion is evaluated by one-versus-one SVM classifier via two metrics: overall accuracy and mean average precision (mAP). The overview of the proposed architecture is illustrated in Fig. 3.

Table 1 Performance of individual CNN models vs. proposed fusions on RGB

Models	Stanford 40		PPMI	
	Acc (%)	mAP (%)	Acc (%)	mAP (%)
V	76.68	75.68	59.31	59.46
G	76.65	75.45	59.46	59.63
R50	79.32	78.20	55.93	55.96
R101	81.72	80.84	61.36	61.41
R152	81.38	80.53	60.36	60.56
(Proposed Fusion Model)				
V+G+R50	81.63	80.59	62.56	62.64
V+G+R50+R101+R152	83.69	82.64	64.94	64.93

6 Experimental Setup, Results, and Discussion

Preliminary benchmark experiments of each CNN model are run on Caffe (11), an open source deep learning software framework developed by the Berkeley Vision and Learning Center. The hardware configuration of our system consists of two NVIDIA GeForce GTX TITAN X GPU with 12GB of VRAM each and two Intel Xeon processors E5-2690 v3 2.60GHz with a total of 48/24 logical/physical cores and 256 GB of main memory.

Experiments are organized in three sets, each toward specific goal. The first set of experiments examines performance of different CNN feature fusions on RGB. The second set considers discriminative power of color fusion between RGB and oRGB for each individual CNN model. The last set applies fusion on all five models and two color spaces. All experiments are conducted on both datasets: Stanford 40 and PPMI.

6.1 CNN Model Fusions

The first set of experiments focuses on examining the late fusions of features extracted from VGG-19 (V), GoogLeNet (G), and three ResNets: ResNet-50 (R50), ResNet-101 (R101), and ResNet-152 (R152) on RGB color space for both Stanford 40 and PPMI datasets. Performances of two model fusions V+G+R50 and V+G+R50+R101+R152 are evaluated against individual CNN models. Performance of each single model and each fusion are measured in overall accuracy and mean average precision and recorded in Table 1. In addition, performance measurements of Stanford 40 dataset are represented in Fig. 4 for clearer visualization of the improvements of fusions over their individual models.

It is obvious to recognize that all model fusions outperform their component CNN models. In particular, both each fusion improves at least 2% overall accuracy and mean average precision over component models on Stanford 40. For PPMI, the improvement of model fusion is more significant as it achieves more than 3% gain in both performance metrics. Although large ResNets such as ResNet-101 and ResNet-152 give equivalent performance compared to the small fusion of ResNet-50 with GoogLeNet and VGG-19, V+G+R50, fusing these robust ResNets with the small fusion V+G+R50 continues to achieve significant improvement over the single robust nets. This result indicates that investigation on late CNN model fusions holds a promising future.

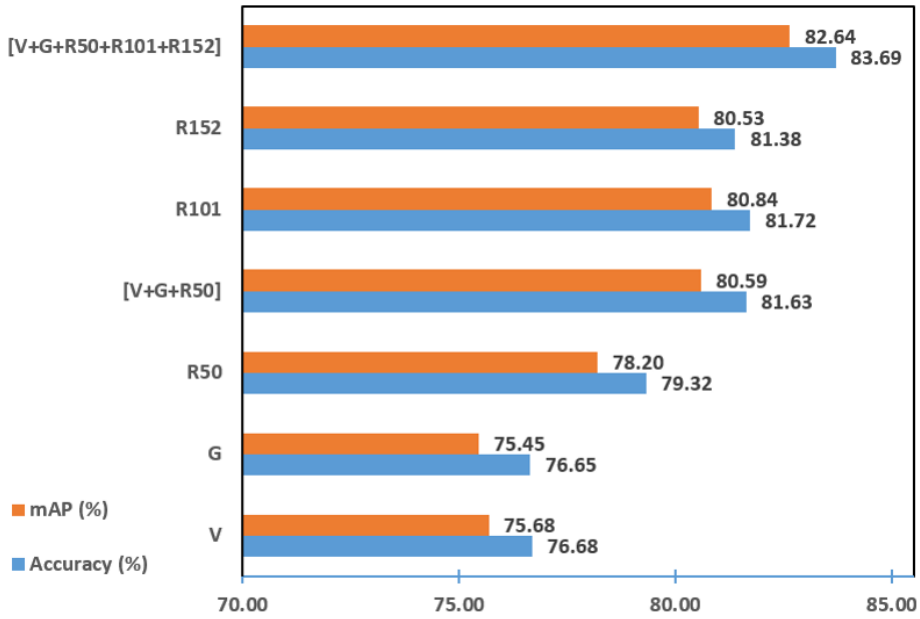

Figure 4 Accuracy(%) and mean Average Precision(%) comparison between individual VGG-19 (V), GoogLeNet (G), ResNet-50 (R50), ResNet-101 (R101), and ResNet-152 (R152) models with proposed fusion models [V + G + R50] and [V + G + R50 + R101 + R152] for Stanford 40 dataset on RGB.

6.2 Color Fusion on CNN Models

Experiments in this section scrutinize late fusions that are created by features of individual CNN models on two color spaces: RGB and oRGB. In other words, a model X will be trained and tested on both color spaces. Extracted features of model X on RGB will be concatenated with those of model X on oRGB to form an RGB+oRGB fusion of model X. Comparison of mAP performance of color fusion of all five CNN models against each individual model on a color space are illustrated in Fig. 5.

One known drawback in this experiment set is that CNN models are trained on oRGB with pre-trained parameters learned on RGB ImageNet dataset. It could be the reason why performance of any model on oRGB is far worse than that on RGB. It may have weakened the performance of color fusion in this experiment set. However, empirical results still indicate the improvements of color fusion over performance of any model except GoogLeNet on RGB for both datasets. For Stanford 40, improvement of color fusion is approximately 0.5% on strong models such as ResNet-101 and ResNet-152 and very little on weak models. However, improvement of color fusion is significant with 1-2% mAP increment on PPMI dataset.

6.3 Evaluation of Color and Model Fusions

In this final set of experiment, we combine the ideas that we have elaborated separately in previous sections: model fusion and color fusion. In this experiment we propose a final fusion of 10 models formed by all five models on two color spaces: RGB and oRGB. Performance of this 10-model fusion will be evaluated against the 5-model fusion on RGB that was

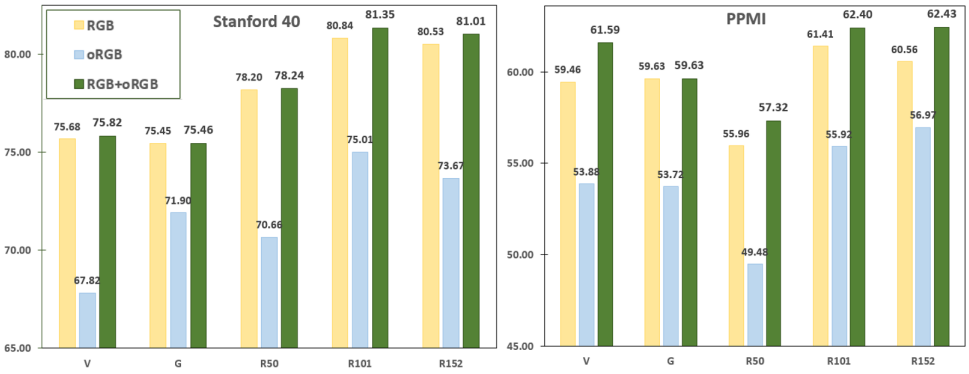

Figure 5 Mean Average Precision(%) comparison between proposed RGB and oRGB color fusion and separate colors for each individual CNN model on two datasets: Stanford 40 (left) and PPMI (right).

Table 2 Comparison with other methods on Stanford 40 (in mAP and overall accuracy)

Model	Accuracy	mAP
Yao and Fei-Fei (6)	-	45.70
Khan *et al.* (22)	-	75.40
Zhao *et al.* (23)	-	78.80
Sharma *et al.* (21)	-	72.30
Zhang *et al.* (26)	-	***82.64***
Rosenfeld and Ullman (VGG-GAP + VGG-16) (29)	81.74	
Rosenfeld and Ullman (VGG-16 + ResNet-152) (20)	***83.12***	
5-model fusion on RGB (Proposed)	83.69	82.64
10-model fusion on RGB and oRGB (Proposed)	**84.24**	**83.25**

examined in the first experiment set and against previous works by other methodologies in Table 2 for Stanford 40 and Table 4 for 24-action PPMI.

6.3.1 Stanford 40

First of all, 10-model fusion on RGB and oRGB surpasses 5-model fusion on RGB by approximately 0.5% for Stanford 40 in both performance metrics. The proposed 10-model fusion outperforms the best individual CNN model, ResNet-101, by approximately 2.5% in both overall accuracy and mean average precision. Our final fusion model surpasses the best overall accuracy in the work that was based on the fusion of VGG-16 and ResNet-152 of Rosenfeld et al. (20) by more than 1.1%. The proposed model also outperforms the current highest mean average precision that was achieved by Zhang et al.(26). This performance gain shows that we could collect more features by using 10 deep CNN models; features that improve discriminatory power but may be overlooked by using individual model or two models.

We also evaluated our model's performance on each Stanford 40 class. Looking at the accuracy column on Table III we see that the highest accuracy is achieved in class "riding a horse" with 97.96. High accuracy in "riding a horse" could be explained by the majority of

the images in the class having full body visibility for both the human and horse, although the poses still vary. Background may also play a role, considering that people usually ride horses in an open field. This significance of background features is the reason we opted to use the whole image and not cropped bounding boxes during preprocessing.

Additionally, the horse's high visibility may add significant clue as well. The importance of aspect can be observed in another class that also involves horses: "feeding a horse." This class has only 94.65% accuracy, which is slightly lower than "riding a horse," although still achieves high accuracy. We think that this is because of the images' lower visibility and occlusions. Some only depict the upper part of the human body and many only show the horse's head as its body is hidden behind fences. The human in many of these images are also occluded.

The class with second highest accuracy is "climbing" with 97.95. Reiterating the significance of visibility, we notice that the majority of images in "climbing" gives a full view of the human body. To generalize further, we observe that visibility on either human body parts or the interactive object can improve or weaken accuracy results as they offer significant cues for the corresponding action. We see that most images in the top 20 classes on Table III have high degree of object visibility with some almost always in full view, which offers important cues in recognizing the action.

Actually, some classes such as "climbing" and "jumping" even require consistent full view of the human body for us to justify the ground truth image label. Showing only the upper part of the human body and labeling that the image belongs to a "jumping" class makes little sense. Furthermore, it also helps that interactive objects of the top 20 classes such as horse, microscope, wall, bike, fishing rod, car, bow, violin, guitar, umbrella, dog, floor, board, TV, cart, and tree are bigger in size, which increases their level of visibility. To put it another way, their relatively bigger size limits their level of occlusion. For instance, we cannot justify the ground truth label of an image as "fishing" if we cannot see much of the fishing rod.

Looking at the lowest of the top 20 classes, "fixing a bike" with 85.94, we think this lower accuracy is due to the images showing partial bike or even only the wheel. The background of many of these images also varies as the action seems to occur in almost random background setting. This, as we observe, can impede classification accuracy.

On another extreme, we observed that many images of the lowest 20 classes have object occlusion. Classes such as "texting message," and "smoking" contain images with high degree of occlusion on the interactive object (phone or cigarette). Due to their relatively small size compared to human hands, being wrapped in a person's fingers covers much of the object. In contrast, objects such as fishing rod and bow, which are also wrapped in a person's fingers are still visible since they are bigger relative to the human hands.

6.3.2 PPMI

The proposed 10-model fusion on RGB and oRGB significantly outperforms the current state-of-the-art with more than 14% gain in mean average precision on PPMI dataset (Table 4). It also surpasses the 5-model fusion on RGB by approximately 1% and surpasses the best single CNN model, ResNet-101 (Table 1) by approximately 4.5% in both performance metrics. Results of experiment set prove the robust improvement of our color and CNN model fusion over the BoW framework and fusion of engineering feature in the prior work (37; 38).

Table 3 Accuracy and mAP for top 20 action classes of Stanford 40 using proposed 10-model fusion on RGB and oRGB

Action	Accuracy	mAP
playing guitar	95.24	97.42
riding a horse	97.96	96.78
feeding a horse	94.65	96.58
playing violin	95.63	95.99
riding a bike	96.89	95.96
holding an umbrella	95.83	95.66
climbing	97.95	95.15
walking the dog	94.30	93.21
cleaning the floor	93.75	92.59
fishing	89.60	92.51
looking through a microscope	93.41	92.46
watching TV	92.68	92.39
writing on a board	92.77	91.21
shooting an arrow	92.11	91.01
rowing a boat	95.29	90.77
jumping	87.69	90.03
fixing a car	96.03	89.98
pushing a cart	87.41	88.55
cutting trees	86.41	88.28
fixing a bike	85.94	87.13
Overall	84.24	83.25

Table 4 Comparison with other methods on 24-action of PPMI

Model	Accuracy	mAP
Zhao *et al.* (CF) (38; 37)		46.70
Sharma *et al.* (Dsal) (39)		49.40
Zhao *et al.* (GSPM) (38)		***51.70***
(Proposed Fusion Model)		
5-model fusion on RGB	64.94	64.93
10-model fusion on RGB and oRGB	**65.94**	**65.85**

Individual action performance of PPMI

For each instrument, playing action performs far better than non-playing gesture according to empirical results for individual action performance of PPMI in Table 5. The first column in the table records action IDs that are labeled by two concatenated IDs: the ID for the instrument and the ID for specific activity with the instrument. Instrument ID is labeled from 1 to 12 to specify 12 instruments. There are only two activity IDs: 0 for playing an instrument and 1 for being with an instrument. Following the action ID convention, it is simple to compare performance of "play" against "with" each instrument and identify that "play" tremendously outperforms "with" for most individual instruments. For some particular

Table 5 Individual action performance on PPMI using proposed 10-model fusion on RGB and oRGB

Action ID	Action	Accuracy	mAP
1.0	play bassoon	83.91	72.45
2.0	play cello	75.27	77.96
3.0	play clarinet	45.05	49.67
4.0	play erhu	82.11	77.91
5.0	play flute	69.32	67.07
6.0	play frenchhorn	67.05	68.52
7.0	play guitar	89.00	86.35
8.0	play harp	84.85	84.37
9.0	play recorder	55.17	55.48
10.0	play saxophone	77.78	70.46
11.0	play trumpet	50.53	50.87
12.0	play violin	84.38	82.64
1.1	with bassoon	40.74	50.98
2.1	with cello	70.79	71.41
3.1	with clarinet	34.18	34.03
4.1	with erhu	59.74	67.71
5.1	with flute	59.38	54.98
6.1	with frenchhorn	73.24	73.69
7.1	with guitar	55.56	66.04
8.1	with harp	86.73	87.04
9.1	with recorder	32.00	36.26
10.1	with saxophone	75.58	77.93
11.1	with trumpet	50.57	55.65
12.1	with violin	57.14	61.01
Overall		65.94	65.85

instrument such as guitar, playing the instrument surpasses being with the instrument by more than 33% in accuracy and 30% in average precision.

As was mentioned in sub-section 5.2, we decided to use full images with background in our methodology to achieve the best performance. The idea of preserving action background is reinforced by the current observation on performance of "play" against "with." Action background is an important factor to explain why playing an instrument usually performs far better than being with an instrument. People usually play an instrument in some specific scenarios, thus, images of "play" actions are captured with sufficiently relevant background. Meanwhile, in "with" actions, instruments can be with people anywhere and the background contributes less in recognizing the actions. Another factor to explain weak performance of "with" instruments is the lack of precise human gesture in those actions. In other words, being with an instrument is a broad concept with a large range of human gestures such as standing, sitting, and holding as long as the instrument exists.

Additionally, the confusion matrix in Fig. 6 further provides a visualization of action performance relation on PPMI, especially with the arrangement of actions by instrument and activity in our work. The color legend indicates that light blue shades represent low percentage and dark blue for high percentage toward 100%. Beside the expected dark

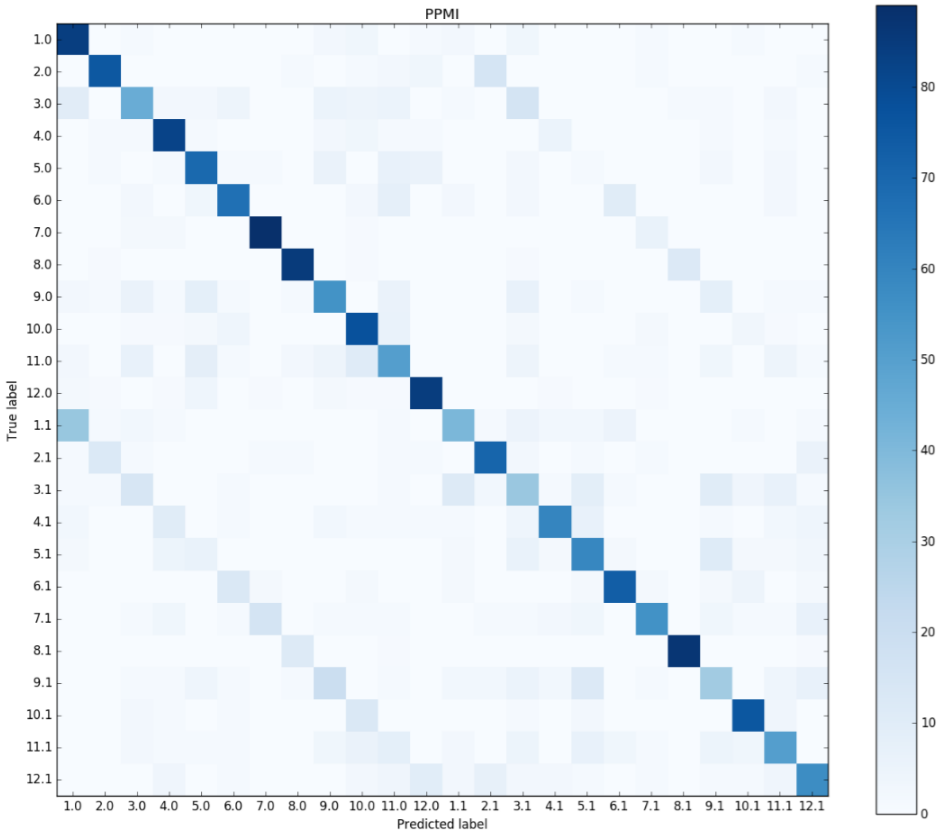

Figure 6 Confusion matrix of proposed 10-model fusion on PPMI dataset.

diagonal that indicates high percentages of correctly classified actions, two mid-diagonals of the bottom left and top right half matrices carry interesting information. The bottom left mid-diagonal shows that playing each particular instrument is commonly misclassified as being with the instrument. The top right mid-diagonal specifies the percentage that "with" an instrument actions are misidentified as playing the instrument. It means that it is hard to accurately identify "play" against "with" and vice versa for each specific instrument.

t-SNE of 10-model vs. ResNet-152

Lastly, t-Distributed Stochastic Neighbor Embedding (t-SNE) (12) is applied to visualize the discrimination of features in the proposed 10-model method. In short, t-SNE is a technique for dimensionality reduction that is well suited for visualization of high-dimensional datasets. Fig. 7 shows t-SNE visualization of CNN features that are extracted from 24-action PPMI test dataset by the proposed 10-model method and the most complex individual CNN model, ResNet-152, side by side. From this image we can see that the proposed 10-model method generates a more distinctive separation of action clusters such as play harp against with harp, play cello against with cello, and play/with guitar against others compared to those of the individual ResNet-152.

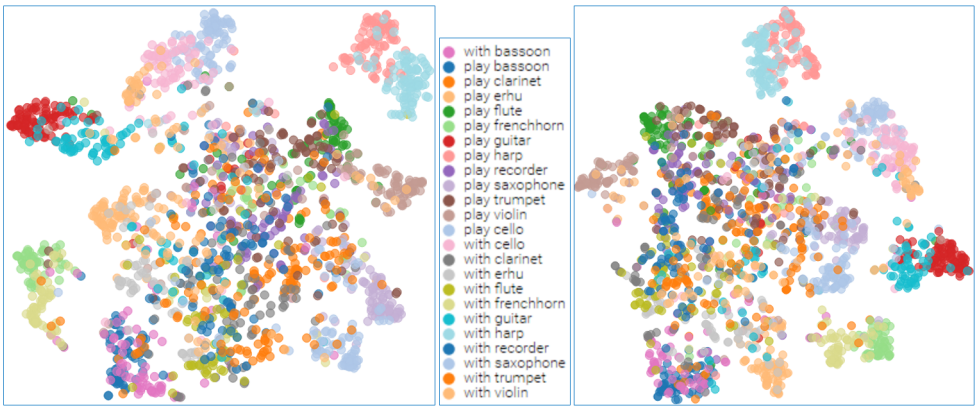

Figure 7 t-SNE (12) visualization of proposed 10-model fusion (left) vs ResNet152 (right) on PPMI

7 Conclusion and Future Work

We have examined the fusions of high performing deep CNN models and two color spaces to prove the performance improvements of fusions by CNN models, by color space, and by the combination of both CNN models and color spaces. We have proposed a final 10-model fusion method that are built upon five CNN models: GoogLeNet, VGG-19, ResNet-50, ResNet-101, and ResNet-152 and two color spaces: RGB and oRGB. Our final fusion model achieves a gain of more than 2.5% and 4.5% on Stanford 40 and PPMI respectively over the best single CNN. This improvement also boosts our proposed methodology performance beyond other prior works for both mentioned datasets.

In addition, we have emphasized the role of background in identifying an action. In the future work, we can advance our fusion technique to combine features from full image with those from bounding boxes. Proposed algorithm provides an effective paradigm that can be applied to a broad range of applications such as police body worn camera, security surveillance, and neighborhood watch program.

References

[1] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 2-16, 2010.

[2] J. Lai et al., "Image-based vehicle tracking and classification on the highway," *Int. Conf. on Green Circuits and Systems (ICGCS)*, Shanghai, China, 2010.

[3] R. Maree et al., "Biomedical image classification with random subwindows and decision trees," *ICCV Workshop on Computer Vision for Biomedical Image Applications*, Beijing, China, 2005.

[4] T. Brosnan and D. Sun, "Improving quality inspection of food products by computer vision - a review," *J. Food Engineering*, vol. 61, no. 1, pp. 3-16, 2004.

[5] D. Kim et al., "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," *Int. Conf. on Robotics and Automation (ICRA)*, Orlando, FL, 2006.

[6] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," *Int. Conf. on Comp. Vision and Pattern Recognition (CVPR)*, San Franciso, CA, 2010.

[7] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. on Computer Vision*, 2015.

[8] A. Krizhevsky et al.,"ImageNet classification with deep convolutional neural networks," *Annu. Conf. on Neural Information Processing Systems (NIPS)*, Lake Tahoe, 2012.

[9] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, "Visual categorization with bags of keypoints", *8th European Conference on Computer Vision (ECCV)*, 2004.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004.

[11] Y. Jia et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.

[12] L.J.P. van der Maaten, "Accelerating t-SNE using Tree-Based Algorithms," *J. of Machine Learning Research*, vol. 15, pp. 3221-3245, Oct. 2014.

[13] C. Szegedy et al., "Going deeper with convolutions," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015.

[14] S. Arora et al., "Provable bounds for learning some deep representations," *Int. Conf. on Machine Learning (ICML)*, Beijing, China, 2014.

[15] M. Lin, C. Qiang, and Y. Shuicheng, "Network in network," arXiv preprint arXiv:1312.4400, 2013.

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

[17] K. He et al., "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385, 2015.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2014.

[19] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *British Machine Vision Conference (BMVC)*, Nottingham, United Kingdom, 2014.

[20] A. Rosenfeld, S. Ullman, "Action Classification via Concepts and Attributes," *arXiv preprint arXiv:1605.07824v1*, 2016.

[21] G. Sharma, F. Jurie, C. Schmid, “Expanded Parts Model for Semantic Description of Humans in Still Images", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, issue 1, 2017.

[22] F. S. Khan, J. Xu, J. van de Weijer, A. D. Bagdanov, R. M. Anwer, A. M. Lopez, “Recognizing Actions Through Action-Specific Person Detection", *IEEE Trans. Image Process.*, vol. 24, no. 11, 2015

[23] Z. Zhao, H. Ma, X. Chen, “Multi-scale Region Candidate Combination for Action Recognition", *IEEE International Conference on Image Processing (ICIP)*, 2016.

[24] N. Shapovalova, W. Gong, M. Pedersoli, F. X. Roca, J. Gonzalez, “On importance of interactions and context in human action recognition", *Pattern Recognition and Image Analysis: 5th Iberian Conference*, vol. 6669, p. 58, 2011.

[25] F. Sener, C. Bas, N. Ikizler-Cinbis, “On recognizing actions in still images via multiple features", *European Conference on Computer Vision Workshops and Demonstrations (ECCV)*, pp. 263-272, 2012.

[26] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu, “Action Recognition in Still Images with Minimum Annotation Efforts," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5479-5490, 2016.

[27] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154-171, 2013.

[28] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117âŁ“128, 2011.

[29] A. Rosenfeld, S. Ullman, “Visual Concept Recognition and Localization via Iterative Introspection", *arXiv preprint arXiv1603.04186v2*, 2016

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization," *arXiv preprint arXiv:1512.04150*, 2015.

[31] H. He and E. A. Garcia, “Learning from imbalanced data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 21, no. 9, pp 1263-1284, 2009.

[32] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Int. Conf. on Machine Learning (ICML)*, Lille, France, 2015.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “ Identity mappings in deep residual networks," arXiv:1603.05027v2, 2016.

[34] B. Yao and L. Fei-Fei, “Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[35] S. Banerji, A. Verma, and C. Liu. “Novel color LBP descriptors for scene and image texture classification." *15th International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2011.

[36] M. Bratkova, S. Boulos, and P. Shirley. “oRGB: a practical opponent color space for computer graphics", *IEEE Computer Graphics and Applications*, vol 29, no.1, pp 42-55, 2009.

[37] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, A. M. Lopez, M. Felsberg, “Coloring action recognition in still images", *International Journal in Computer Vision (IJCV)*, vol. 105, no. 3, pp 205-221, 2013

[38] Zhao, Zhichen, Huimin Ma, and Xiaozhi Chen. "Generalized Symmetric Pair Model for Action Classification in Still Images." Pattern Recognition (2016).

[39] G. Sharma, F. Jurie, C. Schmid, “Discriminative spatial saliency for image classification", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2012.