Fusion Based Deep CNN for Improved Large-Scale Image Action Recognition

Yukhe Lavinia[*], *Holly H. Vo*[*], *and Abhishek Verma*
Department of Computer Science
California State University
Fullerton, California 92834, USA
Email: {ylavinia, hhvo}@)csu.fullerton.edu, averma@fullerton.edu

Abstract**—** Still image-based action recognition is a process of labeling actions captured in still images. We propose a fusion method that concatenates two and three deep convolutional neural networks (CNN). After examining the classification accuracy of each deep CNN candidates, we inserted a 100 dimensional fully-connected layer and extracted features from the new 100 dimensional and the last fully-connected layers to create a pool of candidate layers. We form our fusion models by concatenating two or three layers from this pool—one from each model—and trained and tested them on the large-scale Stanford 40 Actions still image dataset. We forwarded the concatenated features to Random Forest and SVM for classification.

Our experiments show that our fusion of two deep CNN models achieved better accuracy than the individual models, with the best performing fusion duo of 80.351% accuracy. The fusion of three models increases the accuracy even further, performing better than both the individual and the fusion of two models, with 81.146% accuracy. Moreover, we also investigate the classification difficulty level of the Stanford 40 Actions category.

Keywords- deep convolutional neural networks; deep learning fusion model; action recognition; VGGNet; GoogLeNet; ResNet;

I. INTRODUCTION

Over the past years, machine learning in computer vision has gained significant advances in solving multitudes of image classification problems. These successes have led to applications in various domains: remote sensing [1], traffic and vehicle surveillance [2], biomedical image classification [3], food product quality inspection [4], and robot navigation [5]. Action recognition is one area that profits from image classification advances as many have used image classification methods to solve action recognition problems [6].

Action recognition can be loosely defined as labeling a video or still image with "verbs" that portray the corresponding action contained in that video or image. Broader applications of action recognition include security surveillance [7], elder and child-care monitoring [8], and human-computer interaction.

Action recognition methods are developed based on its action representation medium. Video-based action recognition uses image sequences as input, while still image action recognition uses still images as input. In recent years still image-based action recognition has gained popularity [9].

A few still image-based action recognition methods focusing on image representations are Visual Concepts and Object Bank. These methods seek to optimize features acquisition upon which the classifiers would be trained. Visual Concepts focuses on mid-level representations and makes use of text queries on popular search engines to harvest visual attributes of the images throughout the internet [10]. Learning is done from these collected visual concepts. Object Bank is a high-level image representation formed from the response maps of multiple object detectors [11]. Some methods take the following cues to improve action recognition performance: human-object interaction [12], human pose [13], and body parts [14].

Focus on advancing still image-based action recognition benefits other research topics related to it. A more accurate still image-based action recognition could be used to reduce number of frames in video-based action recognition. Also, accurate action recognition would give more reliable cues in pose estimation, image retrieval, scene recognition, and object detection. Similarly, we expect that advances in these topics also benefit still image-based action recognition.

The annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [27] has proven to be a popular arena where image classification and object detection researchers present and compare their algorithms. Since the revolutionary AlexNet [15], many teams that performed well on the ILSVRC used a type of deep CNN: VGGNet [16], GoogLeNet [17], and ResNet [18].

Our goal is to improve action recognition accuracy on still images. Considering the increased performance of deep CNN in solving visual recognition problems that are related to action recognition, we expected that VGGNet, GoogLeNet, and ResNet would also perform well on action recognition.

In this paper, we propose and evaluate the following:

(1) Fusion of two deep CNNs achieves higher action recognition accuracy than individual models
(2) Fusion of three deep CNNs achieves an even higher accuracy than the two-CNN fusion

Additionally, we also used our best performing fusion model to investigate the dataset's difficulty level for action classification.

The rest of the paper is organized as follows: section II gives a brief overview on the deep CNN that we fused, section III describes the dataset, section IV elaborates on the methodology, section V describes the experimental results. Section VI discusses conclusion and future work.

[*]these authors have contributed equally to this work

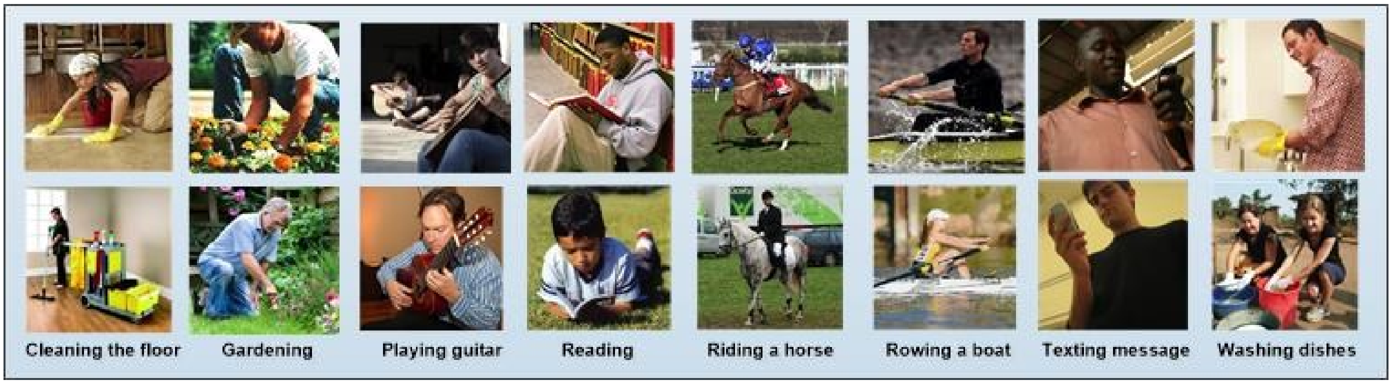

Fig. 1. Example images of Stanford 40 Actions dataset.

II. Overview of Recent Deep Learning Models

Our method uses deep CNNs as the base of our fusion model. The deep CNN models we fused are GoogLeNet, VGG-19, and ResNet-50. The following describes the architecture of each model.

GoogLeNet. Inception-v1 [17] or as it is commonly called "GoogLeNet," is a deep yet lightweight network as its primary idea is performance improvement and computational efficiency. GoogLeNet's relatively low computational cost is a product of two ideas: 1) optimal convolutional neural network through the use of sparsity [19]; and 2) dimensionality reduction through 1x1 convolutional layer as proposed by Lin *et al.* in their model Network-in-Network (NIN) [20].

The Inception-v1 modules in GoogLeNet use 1x1, 3x3, and 5x5 filters and also a max-pooling layer, all arranged in parallel fashion. To reduce dimensionality, the 3x3 and 5x5 filters are preceded by a 1x1 convolutional layer, while the max-pooling layer is succeeded by a 1x1 convolutional layer. The overall GoogLeNet architecture is wide (due to its parallel-fashioned Inception-v1 modules) and deep (22 layers) yet computationally efficient, mainly due to its first few layers being traditional convolutional layers and the rigorous dimensionality reduction using the 1x1 convolutional layers.

VGGNet. This deep network of 16 or 19 layers employs small (3x3) convolutional layers with stride 1 in the whole network [16]. The purpose is twofold: 1) to increase the discriminative power of the rectified linear activation by having more layers (two or three layers as opposed to one, as in the case of 7x7 receptive field), and 2) to decrease the number of parameters. Followed by three fully-connected layers, VGGNet is a computational heavyweight but highly performing network.

ResNet. One of the deepest neural networks, if not the deepest with its current incarnation of 1,001 layers [21], residual network [18] employs shortcut paths that perform identity mapping in order to achieve the desired output. For each residual unit, an input is branched: one goes into the function and is transformed (the "residual") while another bypasses the function (the "identity"). By merely adding the "residual" to the "identity," optimization becomes easier since that means we don't have to generate the entire desired output through conventional method.

The residual network implementation employs batch normalization (BN) to reduce internal covariate shift [22]. For deep neural network such as residual network, a change in activations and input distribution in early epochs could cause great changes in the later epochs, which will affect accuracy. As deep neural network utilizes stochastic gradient descent, BN solves the problem by normalizing each mini-batch.

III. Description of Stanford 40 Action Dataset

Compared to video, still images cost less memory footprint. The efficacy of still image in capturing the desired action, however, depends on the actions. Some are more suitable to be represented as still image, while others as video. Actions such as sleeping, reading, or running could still be accurately perceived when represented in a still image. On the other hand, performing certain dance moves, making certain hand gestures, or doing other sequential motions would lose their meaning when being represented in a still image. These actions require image sequences to be perceived correctly and thus video would be a better representation medium.

We evaluated our model on the Stanford 40 Actions dataset [23]. Overall, the dataset contains 9,532 still images and has 40 action classes depicting daily activities such as "cooking", "using a computer", "writing on a book", and so on. The images are in various visibility, occlusion, angles, poses, and background clutter. Fig. 1 depicts example images of the Stanford 40 Actions.

Stanford 40 Actions is not the only still image action recognition dataset with various visibility, angles, and background clutter. Ikizler [24] and PASCAL [25] datasets also have those difficult images. However, these two datasets have much lower number of classes, with Ikizler having only 5 action classes and PASCAL having 9. Considering the various visibility, angles, background clutter, and also the number of action classes, Stanford 40 Actions is easily one of the most challenging still image-based action recognition datasets.

Originally, the dataset is divided into 4,000 training images and 5,532 test images. For our experiments, however, we created a validation dataset from the train dataset. Thus, from

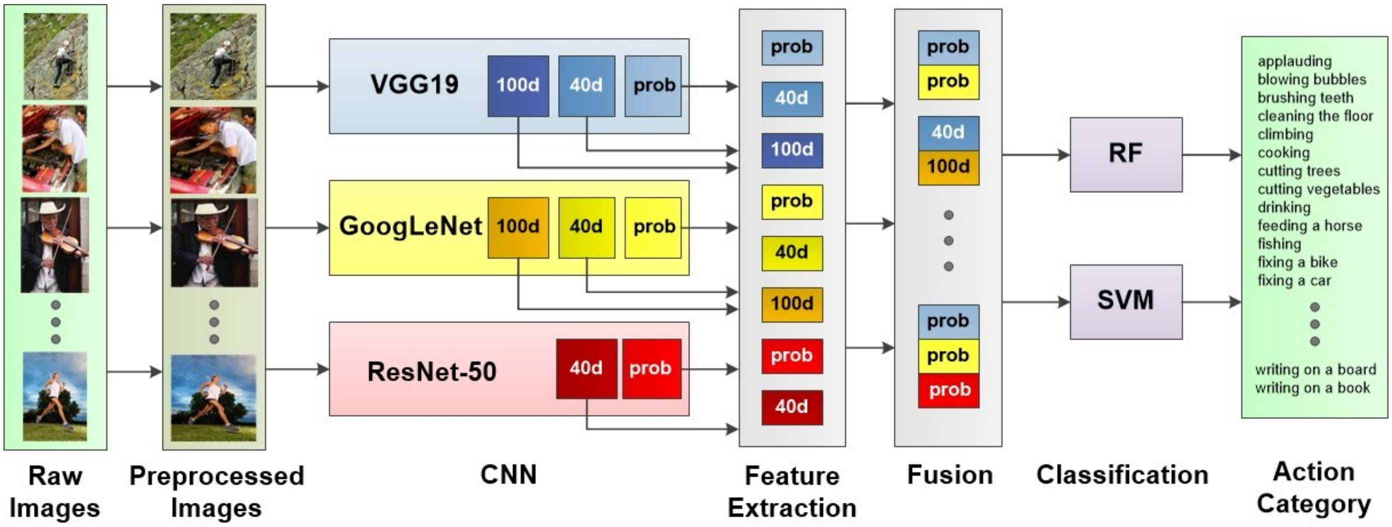

Fig. 2. Fusion methodology. The raw images are preprocessed and forwarded into each deep CNN models: VGG-19, GoogLeNet, and ResNet-50. Feature extraction is done on the 40-d fc layer and prob layer on the three models and also 100-d fc layer on VGG-19 and GoogLeNet. These vectors are concatenated in groups of two and three and forwarded into Random Forest (RF) and SVM classifiers.

the original 4,000 training images, we took 3,200 images to make up our training dataset with 80 images per action class, while the remaining 800 images make up the validation dataset with 20 images per action class. We used the test dataset of 5,532 images.

IV. Proposed Deep CNN Fusion Methodology

We developed our fusion methodology as we experimented on the individual deep CNN models. In this preliminary step, we benchmarked GoogLeNet, ResNet-50, VGG-16, and VGG-19 on the Stanford 40 Actions to gauge each model's action recognition accuracy. We only needed one VGGNet in our fusion model thus after benchmarking the models we chose VGG-19 over VGG-16 due to the former's slightly better performance on our dataset.

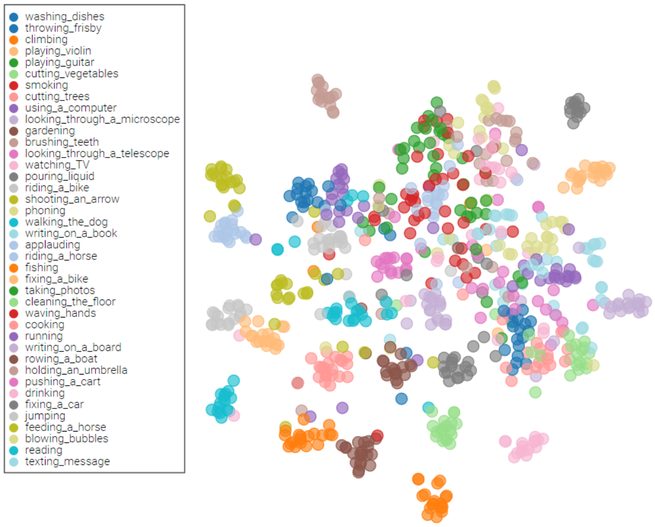

Fig. 3. t-SNE [26] visualization of the Stanford 40 Actions validation set. Clustering results are generated by extracting 40-dimensional features from fully connected layer of ResNet-50.

For image preprocessing, since the raw images come in various sizes, we uniformed them by squashing the images into 256x256, which we later cropped to 224x224. Although the dataset is considered the biggest still image action recognition dataset, we deemed the dataset to be too small to pre-train the models. Thus, we used the models' ImageNet pre-trained weights to fine-tune and test them on the Stanford 40 Actions.

We inserted a 100-dimensional (d) fully-connected (fc) layer before the last 40-d fc layer on the GoogLeNet and VGG-19. The rationale is to take advantage of a higher dimensional feature vectors that supposedly lead to a better performance. Another layer that we used is the prob layer, which is the softmax layer in the deep CNN. The output of this prob layer is a probability vector for which we were interested in extracting to be fused with other models' feature vectors. Extracted features from the 100-d and 40-d fc layers were forwarded to Random Forest and linear SVM. This classification step is necessary to determine the best performing layer on each model. These layers are ones that we prioritized to fuse and fine-tune.

Since we experimented with GoogLeNet and VGG-19 first, we found that the prob layers consistently generated better results than the 100-d and 40-d fc layers. Moreover, we figured that running unmodified ResNet-50 was already computationally taxing. Thus, we did not insert a 100-d fc layer on our ResNet-50 model. We found that the unmodified individual ResNet-50 model still achieved the highest accuracy on our dataset. Now that we had the necessary information, we solidified our fusion methodology as shown in Fig. 2.

As in the preliminary experiments, our fusion process started with image preprocessing. Next, we forwarded these preprocessed images into separate deep CNN models. We extracted features from the 100-d, 40-d, and prob layer of VGG-19 and GoogLeNet, and the 40-d and prob layer of ResNet-50. Our fusion models are the possible combinations of these vectors. We concatenated these possible combinations and classified them using Random Forest and linear SVM.

TABLE I. TOTAL AVERAGE ACCURACY OF INDIVIDUAL MODELS (%)

VGG-19	GoogLeNet	ResNet-50
prob	prob	40-d fc
76.464	76.591	**79.049**

In Fig. 3, the "100d", "40d", and "prob" on the CNN step refer to the layers from which we extracted features. The concatenated features are classified using Random Forest (RF) and SVM.

V. EXPERIMENTAL SETUP, RESULTS, AND DISCUSSION

A. *Setup and Implementation*

We ran the experiments on the individual model using Caffe [28], which is an open source deep learning software framework developed by the Berkley Vision and Learning Center. Caffe plugs in into the NVIDIA DIGITS platform [29], which is a Deep Learning GPU Training System. NVIDIA DIGITS [28] is an open source project that enables the users to design and test their neural networks for image category classification and object detection with real-time visualization.

The hardware configuration of our system is one NVIDIA GeForce GTX TITAN X GPU with 12GB of VRAM. The system has two Intel Xeon processors E5-2690 v3 2.60GHz with a total of 48/24 logical/physical cores and 256 GB of main memory.

We trained our VGG-19 model using batch size of 40, base learning rate of 0.0002, gamma of 0.96, momentum of 0.9, and weight decay of 0.0005 for 50 epochs. Our GoogLeNet was trained using batch size of 40, base learning rate of 0.0009, gamma of 0.1, momentum of 0.9, and weight decay of 0.0005 for 30 epochs. Our ResNet-50 model was trained using batch size of 16, base learning rate of 0.0009, gamma of 0.1, momentum of 0.9, and weight decay of 0.0005 for 30 epochs. We used the scikit-learn Python library to implement and test our fusion model using Random Forest and SVM.

We used accuracy to measure our results instead of mean average precision (mAP) since mAP is more suitable for information retrieval while we are interested in finding classification performance.

B. *Experimental Results and Discussion*

We use t-Distributed Stochastic Neighbor Embedding (t-SNE) [26] for visualization of results. t-SNE is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. Fig. 3 shows t-SNE visualization of the Stanford 40 Actions validation set. Clustering results are generated by extracting 40-dimensional features from fully connected layer of ResNet-50.

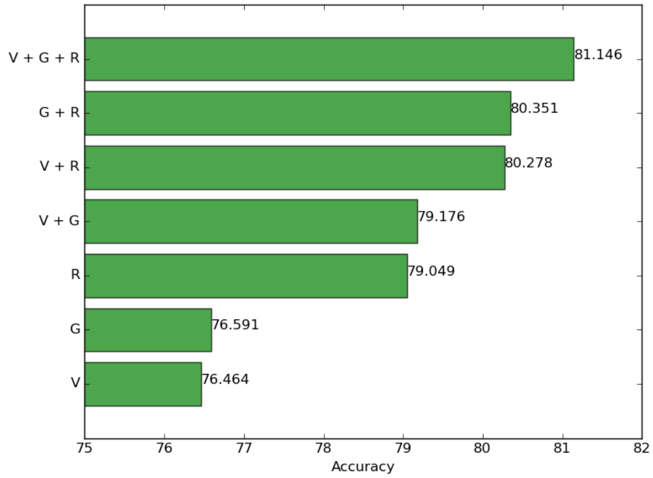

Fig. 4. Accuracy (%) comparison between individual VGG-19 (V), GoogLeNet (G), and ResNet-50 (R) models with fusion models V + G, V + R, G + R, and V + G + R. The values listed here are the same bolded values found in Table II.

Table I lists the best results of our individual deep CNN models. We see that ResNet-50 achieved the highest accuracy with 79.049%. This result was obtained on the 40-d layer using SVM. GoogLeNet and VGG-19 delivered almost similar accuracy, with GoogLeNet getting slightly higher result. Both results were obtained on the prob layer using softmax.

Table II displays the action recognition accuracy of our various fusion models. We only displayed the top 3 layer combinations of each fusion model as there isn't enough space to display all of the layer combination test results. The "Layer" row indicates these best performing layer combination in no particular order. "P" refers to the probability vectors extracted from the models' prob layer while "40" refers to the 40-d layer and "100" refers to the 100-d layer. Thus, "V40", "G40", and "R40" refer to the features extracted from the 40-d layers of the VGG-19 (V), GoogLeNet (G), and ResNet-50 (R) models, and the same naming convention applies to the rest. We displayed the test accuracy results of these fusion models using Random Forest (RF) and SVM. The bolded values are the highest of these classification results.

As we scan the fusion results on Table II, we see that the highest layer combination accuracy increases from 79.176% (VP+GP), 80.278% (VP+R40), 80.351% (G40+R40), and finally 81.146% (VP+GP+R40); all using SVM. Note that this highest result of 81.146% is a fusion the VGG-19 and GoogLeNet's prob layers (VP and GP) and ResNet-50's 40-d fc layer (R40). If we look at Table I, VP, GP, and R40 are the top performing layers in the individual run. VP and GP's contribution can also be seen in V+G, with VP and GP achieving the best results for VGG-19 and GoogLeNet fusion.

We can see that VP shows up numerously in Table II;

TABLE II. TOTAL AVEARAGE ACCURACY OF FUSION MODELS (%)

Model	V+G			V+R			G+R			V+G+R		
Layer	*VP+GP*	*V40+G40*	*VP+G40*	*VP+R40*	*V40+R40*	*V100+RP*	*GP+RP*	*G40+R40*	*G100+R40*	*VP+GP+R40*	*VP+GP+RP*	*VP+G40+R40*
RF	77.657	75.868	76.988	77.585	79.013	79.718	80.152	79.158	79.338	79.48	80.206	78.417
SVM	**79.176**	78.182	78.362	**80.278**	78.688	76.428	80.116	**80.351**	79.827	**81.146**	80.369	80.947

TABLE III. CLASS ACCURACY OF THE V+G+R FUSION MODEL (%)

Class	Result	Class	Result	Class	Result	Class	Result
applauding	71.429	fishing	92.353	playing violin	94.156	taking photos	62.617
blowing bubbles	86.131	fixing a bike	86.822	pouring liquid	59.770	texting message	47.000
brushing teeth	65.179	fixing a car	85.143	pushing a cart	86.861	throwing frisby	76.147
cleaning the floor	84.000	gardening	75.221	reading	60.00	using a computer	77.206
climbing	94.845	holding an umbrella	94.872	riding a bike	93.434	walking the dog	87.562
cooking	83.815	jumping	86.022	riding a horse	95.455	washing dishes	63.158
cutting trees	79.661	looking through a microscope	86.735	rowing a boat	86.316	watching TV	88.525
cutting vegetables	66.667	looking through a telescope	81.818	running	78.431	waving hands	49.580
drinking	75.000	phoning	60.135	shooting an arrow	95.876	writing on a board	85.882
feeding a horse	**96.703**	playing guitar	96.257	smoking	62.016	writing on a book	77.519

VP+R40 generates the highest accuracy for the V+R fusion, VP+GP+RP generating above 80% accuracy results using both RF and SVM; and finally VP+G40+R40 achieving the second best result of all fusion models with 80.947% and VP+GP achieving the highest results in V+G fusion model.

Besides its combination with VP, R40, and RP in the V+G+R and V+G fusion models, GP also contributes in bringing above 80% accuracy in the GP+RP model. GP's contribution is rivaled by G40. For the V+G model, G40 and VP produce one of top 3 best results. For the G+R model, a fusion of G40 and R40 generate the highest accuracy for that model with 80.351%. For V+G+R model, G40 concatenated with VP and R40 also produces one of the top 3 accuracy results.

Another significant layer is R40. As mentioned above, the best performing fusion model is the product of VP, GP, and R40. ResNet's prob layer, RP, is also one of the best contributors in the fusion models.

Fusion of two lower performing individual models (V+G) turned out to be the lowest performing fusion model with 79.176% accuracy. V+R slightly performed better with 80.278%, while G+R 80.351%. The best result is achieved by fusing all three (V+G+R) with 81.146% accuracy.

The results show that the fusion models outperformed the individual models. Even the lowest performing fusion model

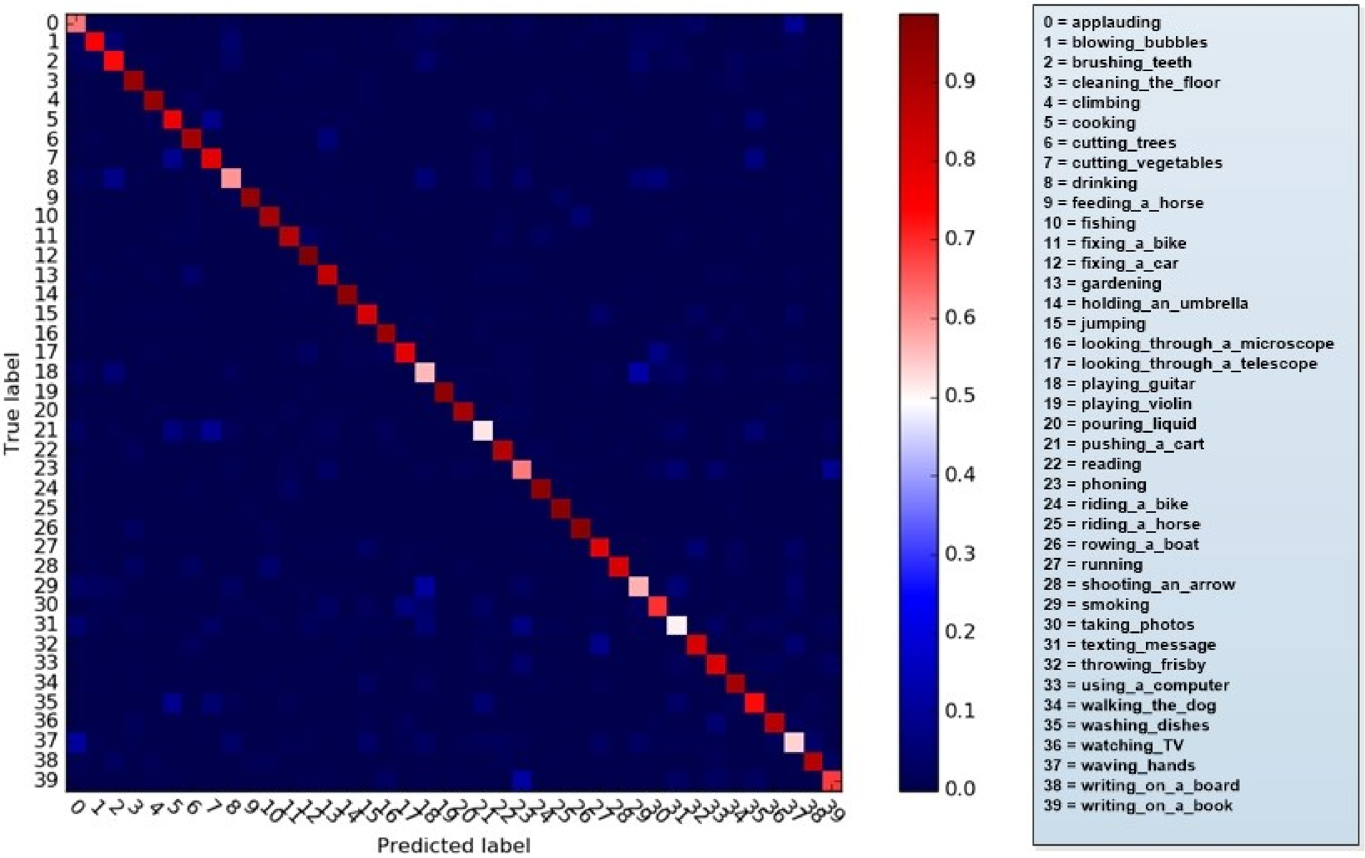

Fig. 5. Confusion matrix of V+G+R.

with two networks (V+G with 79.176%) performed better than the best performing individual model (ResNet's 79.049%). These results strengthen our proposal that fusing two models increased the classification performance.

To further illustrate the fusion models' accuracy in comparison with individual models, Fig. 4 depicts the accuracy results of the three individual models (V, G, and R), the fusion duos (V+G, V+R, and G+R), and the fusion trio (V+G+R).

It shows how different ResNet makes into improving the accuracy. The difference between individual Resnet results and VGG-19 is 2.585%, while it is slightly lower with 2.458% for GoogLeNet. Also, note that the V+G result is 79.176 but the ResNet addition increases the accuracy to 81.146%, which makes a 2.097% difference. This iterates our claim that adding another model to a two-model fusion increases classification accuracy.

Another results that we analyzed are the fusion model's class accuracy (Table III). This refers to the model's accuracy in recognizing certain class. We can also see it as measuring the level of difficulty in recognizing the action categories. The results are taken by using our best performing fusion model, V+G+R. The easiest class to classify, in which the model achieved highest accuracy, is "feeding a horse" with 96.703%, while the most difficult one, which is the lowest class accuracy, is "texting message" with 47%. The confusion matrix in Fig. 5 provides another view to the V+G+R action recognition performance.

VI. Conclusion And Future Work

We have proposed that the fusion of two high performing deep CNN models achieved better action recognition accuracy than an individual model, and that the fusion of three models increased the performance further. Our experiments demonstrated that the fusion of two deep CNNs generated about 2% increase in accuracy, and that adding another powerful deep CNN model to the fusion duo increased another 2% accuracy. We have also investigated the difficulty level of the action categories.

Further investigation could still be done to see if adding another high-performing model to the fusion trio would improve the accuracy. Another idea to improve accuracy is to include object localization method in the fusion methodology.

VII. 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. Mar´ee 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] N. Ikizler-Cinbis et al., "Learning actions from the web," *Int. Conf. on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[7] B. Yao and L. Fei-Fei, "Grouplet: A structured image representation," *Int. Conf. on Comp. Vision and Pattern Recognition (CVPR)*, San Francisco, CA, 2010.

[8] B. Yao et al., "Combining randomization and discrimination for fine-grained image categorization," *Int. Conf. on Comp. Vision and Pattern Recognition (CVPR), Colorado Springs, CO,* 2011.

[9] V. Delaitre et al., "Recognizing human actions in still images: A study of bag-of-features and partbased representations," *British Machine Vision Conf. (BMVC)*, Wales, UK, 2010.

[10] W. Niu et al., "Human activity detection and recognition for video surveillance," *Int. Conf on Multimedia and Expo (ICME)*, Taipei, Taiwan, 2004.

[11] C. Huang et al., "Human action recognition system for elderly and children care using three stream ConvNet," *Int. Conf. on Orange Technologies (ICOT)*, Hong Kong, 2015.

[12] G. Guo and L. A., "A survey on still image based human action recognition," *Pattern Recognition,* vol. 47, pp. 3343-3361, 2014.

[13] Q. Li et al., "Harvesting mid-level visual concepts," *Int. Conf. on Comp. Vision and Pattern Recognition (CVPR),* Portland, OR, 2013.

[14] L.-J. Li et al., "Object bank: A high-level image representation for scene classification and semantic feature sparsification," *Annu. Conf. on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2010.

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

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Int. Conf. on Learning Reprsentation (ICLR)*, 2015.

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

[18] K. He et al., "Deep residual learning for image recognition," *Int. Conf. on Comp. Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016.

[19] S. Arora et al., "Provable bounds for learning some deep representations," *J. Machine Learning Research*, vol. 32, 2014.

[20] M. Lin et al., "Network in network," in *Int. Conf. on Learning Representations (ICLR)*, Banff, Canada, 2014.

[21] K. He et al., "Identity mappings in deep residual networks," *arXiv preprint arXiv:1603.05027*, 2016.

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

[23] 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.

[24] N. Ikizler-Cinbis et al., "Learning actions from the web," *Int. Conf. on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[25] M. Everingham et al., "The PASCAL Visual Object Classes Challenge (VOC2010) Results," 2010.

[26] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Machine Learning Research,* vol. 9, pp. 2579-2605, 2008.

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

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

[29] NVIDIA DIGITS Software. (2015). Retrieved April 23, 2016, from https: //developer.nvidia.com/digits.