Synthetic Image Augmentation for Improved Classification using Generative Adversarial Networks

Keval Doshi

***Abstract*— Object detection and recognition has been an ongoing research topic for a long time in the field of computer vision. Even in robotics, detecting the state of an object by a robot still remains a challenging task. Also, collecting data for each possible state is also not feasible. In this literature, we use a deep convolutional neural network with SVM as a classifier to help with recognizing the state of a cooking object. We also study how a generative adversarial network can be used for synthetic data augmentation and improving the classification accuracy. The main motivation behind this work is to estimate how well a robot could recognize the current state of an object.**

I. INTRODUCTION

In the past few years, there has been significant development in the field of computer vision especially in object recognition. There have been some very effective and popular methods which have been developed such as Support Vector Machines, Convolutional Neural Networks, Artificial Immune System, etc. Among those, Deep learning has been gaining a lot of popularity as it has proved to be one of the most effective techniques for high dimensional datasets such as images. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, which use a variation of multilayer perceptrons designed to require minimal preprocessing. Typical CNNs used for object recognition use dozens of layers which builds thousands of parameters in the network. The key algorithms of CNN can be traced back to the late 1980s. CNNs saw heavy use in the 1990s. It however fell out of fashion with the rise of support vector machines. Interest in CNNs was rekindled again in 2012 when it proved to drastically reduce the error rates and improve classification rates on popular dataset such as MNIST [1], ImageNet [2], etc. However, object state recognition has not been addressed as much even though it has many important applications in robotics, medical, gaming and many more.

One such important application is to teach a robot how to cook, which however involves several important steps one of which is recognizing the state of food items. For example, a robot should be able to recognize the object (e.g. an onion) and identify the state (e.g. diced onion). In robotics, objects at different states would require different manipulation strategies. A significant difficulty with image recognition tasks is collecting generous amount of training data. For example, collecting hundreds of training images for different states of each object might not even be feasible. In this case, synthetic data augmentation needs to be considered. In this paper we analyze a small section of the state recognition problem, which is classification. Classification mainly helps to tell what sate an object is in. We consider Inception v3 as our base model to extract features which are then classified using a Support Vector Machine (SVM) [3] to solve a eleven class image recognition problem. Also, generative adversarial networks (GANs) [4] have been used for synthetic data augmentation. Specifically, we use a CycleGAN [5] for unpaired image-to-image translation. The final model is shown to achieve 81% accuracy. The primary objective of this research is to develop an automated recognition system which:

1) Classifies cooking items according to its present state.
2) Is robust to challenging real-world conditions such as varying video resolution and lighting conditions.

The outline of the current study is as follows: We look at some of the related works in Section II. In Section III, a brief description of the dataset along with different data augmentation methods used is provided. The fourth section consists of an overview of the proposed approach to classification is provided. This section will highlight the computer vision algorithms selected for the purposes of this study. The training and fine-tuning of the algorithm will also be discussed. In the fifth section, the final results along with a discussion of the proposed model as compared to different models is provided. Lastly, concluding remarks, recommendations, and additional research needs for future 154 are presented in the sixth section.

II. RELATED WORKS

A major breakthrough in using neural networks was made by Yann Le Cun when he proposed an algorithm to train Neural Networks. The innovation [6] was to simplify the architecture and to use the back-propagation algorithm to train the entire system. The approach was successful in performing tasks such as OCR and handwriting recognition. Recently, a hierarchical state space markov model was used to recognize cooking activities using Object based task grouping (OTG) in [7]. In robotics, knowing the object states and recognizing archiving the desired states are very important. Sun [8]–[10] presented a novel object learning approach for robots to understand the objects interaction from human. Particularly, a bayesian network was used to represent the knowledge learned based on which the recognition reliability of objects and human was improved, which would further be

*This work was not supported by any organization

[1]Keval Doshi is with Department of Electrical Engineering, University of South Florida, Tampa, USA. kevaldoshi@mail.usf.edu

TABLE I

PARAMETERS USED FOR DATA AUGMENTATION

Name	Augmentation Factors
Rotation	45
Shear	0.2
Zoom	0.2
Rescale	1/255
Horizontal Flipping	True
Vertical Flipping	True

extrapolated to help the robots to properly understand a pair of objects. Huang [11] focused on the requirements of grasps from the physical interactions in instrument manipulation. In [12], a SVM model was used for feature selection in gait classification of leg length and distal mass. A discussion on synthetic data augmentation using generative adversarial networks for improved liver classification is provided in [13].

III. DATASET AND PREPROCESSING

The data used in this paper includes eleven different classes: creamy paste, diced, floured, grated, juiced, julienne, mixed, other, peeled, sliced and whole with 17 cooking objects(carrot, tomato, pepper, onions, cheese, etc). In our work, dataset version 1.2 of the state recognition challenge from USF RPAL lab was used. The dataset was first annotated to classify an image to one of the eleven classes. It consists of 9309 images out of which 6348 images were used as training data and 1377 images were used as validation data wheras the rest was reserved for testing. However, 6348 images are not sufficient for such a intricate classification task. Hence, several augmentation methods were used to further supplement the dataset such as height shifting, rescaling, rotation as well as vertical and horizontal flipping. The precise augmentation factors used are presented in Table I.

Also, a new promising approach for training a model that synthesizes synthetic images known as Generative Adversarial Networks (GANs) is used. GANs have gained great popularity in the computer vision community and different variations of GANs were recently proposed for generating high quality realistic natural images [14], [15]. Also, there has been an increase in the applications which have applied the GAN framework [16]–[18]. Most studies have employed the image-to-image GAN technique to create label-to-segmentation translation, segmentation to-image translation or medical cross modality translations. Some studies have been inspired by the GAN method for image in painting. In the current study we investigate the applicability of GAN framework to synthesize new training images. In particular, we use a Cycle GAN which is capable of translating an image from domain X to to a target domain Y in the absence of paired examples. The objective is to learn a mapping $G : X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, it is coupled with an inverse mapping $F : Y \rightarrow X$ and a cycle consistency loss is introduced to push $F(G(X))X$ (and vice versa) [5].

Fig. 1. Synthetic Data Augmentation using Cycle GAN.

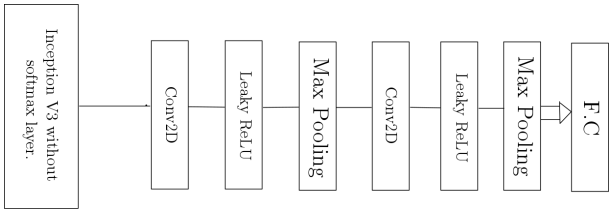

Fig. 2. Modified architecture for Deep Convolutional Neural Network.

IV. METHODOLOGY

The classification system developed in this study consists of a CNN which extracts unique feature descriptors after warping the images to fixed square size of (299 x 299). Then, the feature descriptor for each test image is classified through a linear SVM. The following section explains in detail the procedure followed to build the CNN classifier.

Convolutional Neural Networks are computationally very expensive to train which earlier made them very unpopular for practical applications. However, with the recent rise in efficient graphical processing units along with deep learning libraries such as Tensorflow [19], Torch etc have made CNNs a viable option. In this study, a dual RTX 2070 GPU system was used for training the model. Model training involved two main steps: Base model selection and classifier specific fine tuning.

Base model selection: In real world examples, it might not always be feasible to train a network from scratch. In such cases, we can use a technique known as transfer learning in which a pre-trained model is re-purposed on a second related task. The pre-trained model used in this work is Inception v3, which is a deep convolutional neural network architecture based on GoogLeNet and developed by Google. This architecture achieved the highest accuracy for classification and detection in the ImageNet [2] Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) [20]. Inception performs the utilization of the computational resources in an

improved way with an emphasis on depth and width while maintaining a lower computational complexity.

Modified CNN architecture: As shown in Fig. 2, the base pre-trained model has been modified to better suit our application. First, the last layer of the pre-trained model is removed and a convolution layer is added. A convolution layer generates output in form of tensors as a convolution kernel convoluted with the input of this layer. Then, a batch normalization layer is added, which provides a way to shift inputs closer to zero or mean of all values in the input array. The primary advantage of this technique is to help in faster learning and getting higher accuracy. In a deep convolution neural network, as the data flows further, the weights and parameters adjust their values. This flow can make the intermediate data too big for computation or too small to give correct prediction. However, this problem can be avoided by normalizing the data in mini-batches. Next an activation layer is added with Leaky Rectified Linear Unit (Leaky ReLU) as the activation function. The same layers are added once more before adding a dropout layer, which is used for regularization. Regularization helps the model to generalize better by reducing the risk of over-fitting. There is a risk of over-fitting if the size of data set is too small as compared to the number of parameters needed to be learned. A dropout layer randomly removes some nodes and their connections in the network. The last layer is a fully connected layer.

Specific fine tuning: To adapt the pre-trained model to the proposed task (state recognition), the CNN model parameters are fine-tuned. First, the last softmax classification layer of the pre-trained model is replaced with eleven classes. Stochastic gradient descent is used with a learning rate of 0.001, which allows fine-tuning to make progress while not clobbering the initialization. While training the model, first the entire pre-trained model is frozen and only the last newly added layers are trained for 70 epochs. Then, the first 5 layers of the pre-trained model are fine tuned for 40 epochs to generate the final model. The features extracted are then used to train a SVM model using MATLAB machine learning toolbox.

In the next section, we take a closer look at how different models perform and the accuracy achieved by the best model.

V. Evaluation and Results

As shown in Table II, we performed various experiments to generate the best model. From these experiments, it can be determined that adding more layers does infact help in improving the results. This can be attributed to an increase in the number of parameters because of which the CNN can learn better. The best result was obtained by fine tuning an additional set of layers of the pretrained network, which was Inception V3 in this case. Fine tuning is was done on top of an already trained model, which was the best model considering all previous training examples. The training and validation accuracies and losses are shown in Fig. 3, 4, 5, 6 respectively. However, to further improve the accuracy, SVM with a linear kernel was used. A comparision of different kernels for SVM is provided in Table II. In Table III, we provide a comparision between the performance of different models. The best performance is when two convolutional layers are added along fine-tuning and using SVM as a classifier whereas the worst performance is by Vanilla Inception v3 which does not have any additional layers or fine tuning.

TABLE II
Performance of different SVM kernels

SVM Kernel	Accuracy
Linear	81.3%
Quadratic	79.8%
RBF	79%

We see that linear SVM performs better as compared to SVM with Quadratic or RBF kernel.

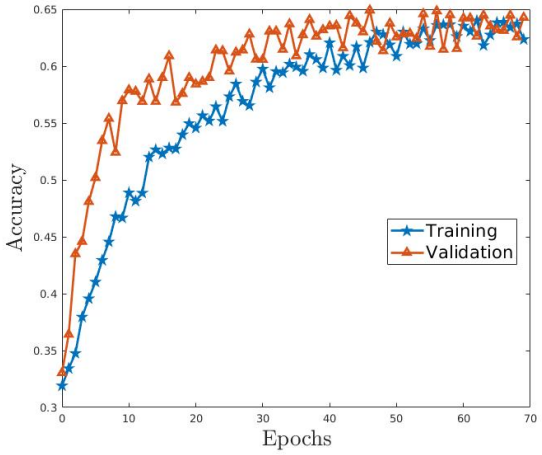

Fig. 3. Accuracy vs Epochs for training and validation during training the newly added layers.

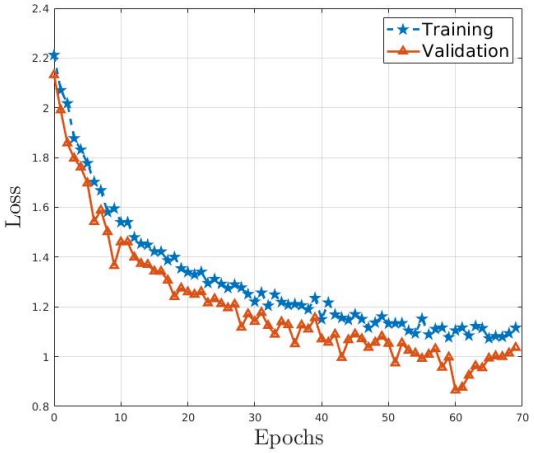

Fig. 4. Loss vs Epochs for training and validation during training the newly added layers.

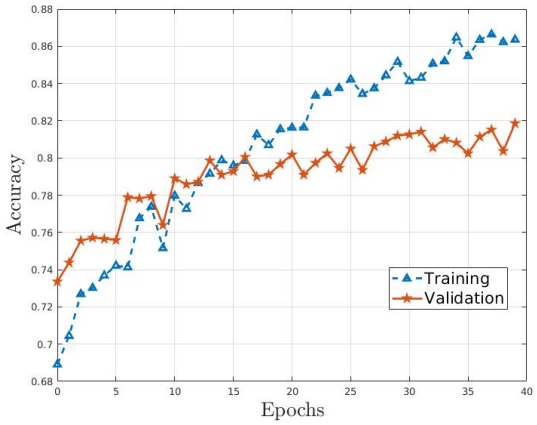

Fig. 5. Accuracy vs Epochs for training and validation during fine-tuning the model.

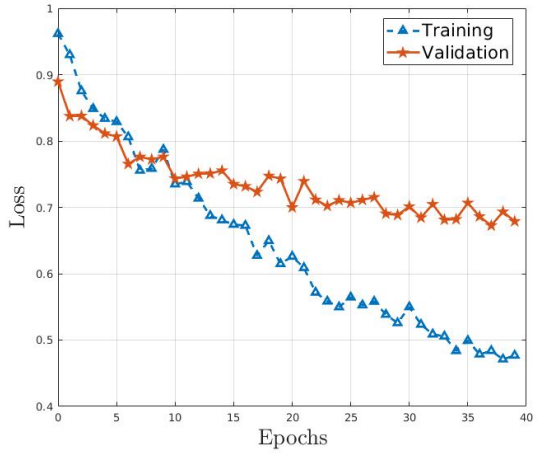

Fig. 6. Loss vs Epochs for training and validation during fine-tuning the model.

VI. Conclusion

In this study we propose a model for detecting the current state of an object. The cooking objects have eleven different states and the best classification accuracy achieved on the test data was 81%. Before this work, there were several studies for object detection but detecting the state of an object has not been researched as much. However, there are a few shortcomings that still need to be addressed for this experiment to viable in a real setting. The training data needs to be properly cleaned as some of the images are have multiple states or are unclear. These images could in turn result in generating a less accurate model. Also, using GAN as way of synthetic data augmentation needs to be explored as a Cycle GAN is only able to output images with size 128x128 with a majority of them being slightly blurry. To implement this in a real life situation a significant amount of study is still required.

TABLE III

Performance of different models on the testing data

Model Trained	Validation Accuracy	Validation Loss	Epochs
Vanilla Inception v3	67%	1.21	70
Inception V3 with one convolutional layer and max pooling.	71%	1.02	70
Inception V3 with two convolutional layers and max pooling.	72%	0.97	70
Inception V3 with two convolutional layers and max pooling along with fine tuning.	76%	0.88	70 + 40
Inception V3 with two convolutional layers, max pooling along with fine tuning and SVM.	81%	0.72	70 + 40

References

[1] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[6] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[7] Prasanth Lade, Narayanan C Krishnan, and Sethuraman Panchanathan. Task prediction in cooking activities using hierarchical state space markov chain and object based task grouping. In *2010 IEEE International Symposium on Multimedia*, pages 284–289. IEEE, 2010.

[8] Yu Sun, Shaogang Ren, and Yun Lin. Object–object interaction affordance learning. *Robotics and Autonomous Systems*, 62(4):487–496, 2014.

[9] Yu Sun and Yun Lin. Modeling paired objects and their interaction. In *New Development in Robot Vision*, pages 73–87. Springer, 2015.

[10] Ahmad Babaeian Jelodar, Ms Sirajus Salekin, and Yu Sun. Identifying object states in cooking-related images. *arXiv preprint arXiv:1805.06956*, 2018.

[11] Yu Sun, Yun Lin, and Yongqiang Huang. Robotic grasping for instrument manipulations. In *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 302–304. IEEE, 2016.

[12] Millicent Schlafly, Yasin Yilmaz, and Kyle B Reed. Feature selection in gait classification of leg length and distal mass. *Informatics in Medicine Unlocked*, page 100163, 2019.

[13] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[15] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[16] Pedro Costa, Adrian Galdran, Maria Inês Meyer, Michael David Abràmoff, Meindert Niemeijer, Ana Maria Mendonça, and Aurélio Campilho. Towards adversarial retinal image synthesis. *arXiv preprint arXiv:1701.08974*, 2017.

[17] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[18] Dong Nie, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 417–425. Springer, 2017.

[19] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[20] Alex Berg, Jia Deng, and L Fei-Fei. Large scale visual recognition challenge (ilsvrc), 2010. *URL http://www. image-net. org/challenges/LSVRC*, 3, 2010.