Pose Recognition via Deep Residual Learning

Deren Lei
Undergraduate
Computer Science
derenlei@ucsb.edu

Zichen Sun
Undergraduate
Computer Science
zichen_sun@umail.ucsb.edu

Abstract

Google's recently released dataset AVA II focusing on activity recognition densely annotates 80 atomic visual actions in 430 15-minute video clips. With its realistic scene and action complexity, AVA II[3] *exposes the intrinsic difficulty of action recognition. The current public focus is on designing time sequence models like Fisher vector encoding to analyze extracted features. Our research project will focus on doing experiments directly on static video frames regardless of the time sequence.*

1. Introduction

The activity labels in AVA II we are interested in are the poses without interactions with subjects. Specifically, actions with labels of stand, sit, walk, run, get up, etc. We will reproduce the residual model, resnet-50 and resnet-101, test them on the MNIST dataset and then doing experiments on the AVA II dataset. The poses without interaction is reasonable to analyzed by static frames for following reasons: 1. There are less consequences since no object will react upon target person's poses. 2. Using static frames will significantly reduce the training time and no need for sequential model. 3. The currently released AVA II dataset does not keep track of each person's position for each frame. Therefore, a separate model for tracking is necessary before analyzing frames as sequences.

2. Data Selection

Despite exciting breakthroughs made over the past few years in classifying and finding objects in images, recognition of human action still remains a big challenge. This is due to the fact that actions are, by nature, less well-defined than objects in videos, making it difficult to construct a finely labeled action video dataset. Actions can be splitted into two categories in action recognition base on their the method of recognition, we name them static recognizable images and dynamic recognizable images. Static recognizable images are easily to recognized via pictures or static images, such as stand, jump and swim, etc. On contrast, dynamic recognizable images are more likely to be recognized in a sequence of actions, such as talking, whirling, pulling, etc. Our model are focusing on static recognizable actions. We compare several datasets including JHMDB and UCF101-24 categories, finally we chose AVA-II datasets which has the lowest mAP (15.6%). Instead of usign TrecVid MED or YouTube-8M which consist of have focused on large-scale video classification, we believe most static recognizable actions should be recognizable with low resolution images. For example, human can easily recognize a people 200 meters away is standing or sitting. Another advantages of AVA is that all the sources of data are movies, which are more realistic and closer to daily life. This means AVA are more diverse and the result of training this dataset would be more widely used, such as school, hospital, etc. We obtain 14 labels of static recognizable actions including stand, sit, walk, bend/bow, lie/sleep dance, run/jpg, crouch/kneel. martial art, get up, jump/leap, fall down, and crawl in AVA-II as the dataset of our model.

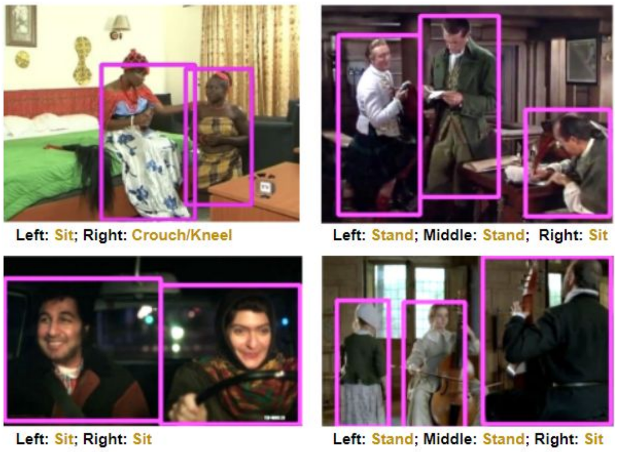

Left: Sit; Right: Crouch/Kneel
Left: Stand; Middle: Stand; Right: Sit
Left: Sit; Right: Sit
Left: Stand; Middle: Stand; Right: Sit

Figure 1. The bounding box and action annotations in sample frames of selected poses from the AVA dataset.

3. Data preprocessing

Data preprocessing contains several steps: uniform aspect ratio and image scaling, image augmentation by random distortion and image normalization.

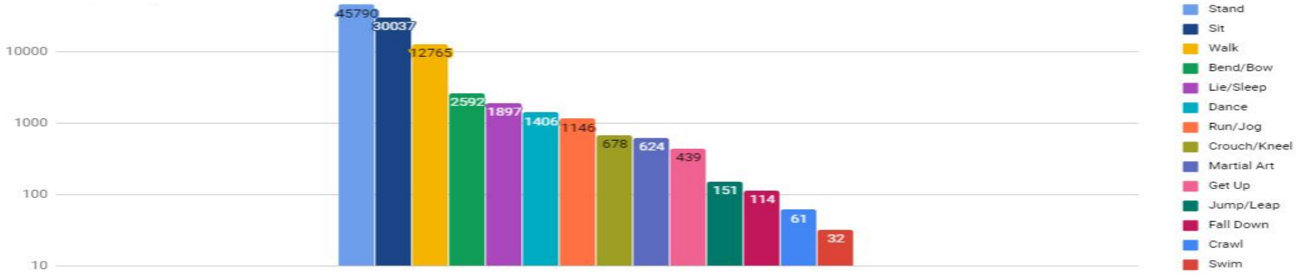


Figure 2. Size of each pose in AVA II training dataset sorted by descending order

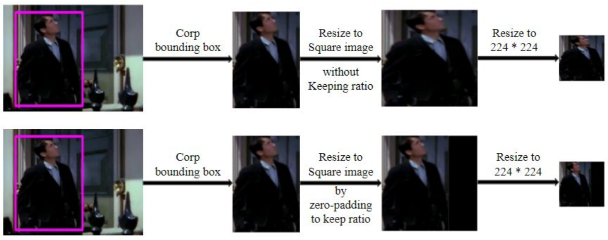


Figure 3. Two options to preprocess the images. Either resize without keeping the ratio or keep ratio by zero-padding.

3.1 Uniform Aspect Ratio and Image Scaling

One of the first steps is to ensure that the images have the same size and aspect ratio. Base on the property of ResNet, the images have to be cropped into 224 by 224 squares. Therefore, two methods of image preprocessing are applied on the dataset. By the first method, the images are first cropped in rectangles and then be stretched into squares and reshape to 224 by 224. Another way of image preprocessing is by zero-padding images into squares and then do reshaping. The trade of between these two method is that that first method would maintain informations in the original images more completely but would loss the ratio of images. Whereas the second method could keep the ratio of images but would add additional information into the data which could influence the quality of images. With zero-padding, the kernels in convolution layers may accidentally learn the zero-padding as features which create some noises for the training process. In addition, the dataset contains some bounding boxes with extreme ratio like 50:1. Thus, We choose not to use zero-padding.

3.2 Image Augmentation by Random Distortion

Image random distortion is common method of image augmentation. First, random distortion operation will certainly increase the numbers of data. In addition, diverse version of a single training input image by random flipping and cropping will increase generalization capability of the model to reduce overfitting.

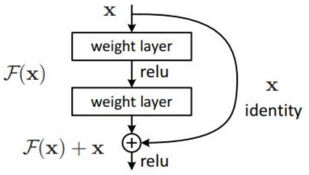

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4. Residual learning: a building block.

3.3 Image Normalization

Image normalization would change the range of image pixel intensity into normal. There are three ways of data normalization that people commonly used:

Data normalization or data scaling:

the data is projected in to a predefined range. This is useful when you have data from different formats (or datasets) and you want to normalize all of them so you can apply the same algorithms over them.

Data standardization:

Data standardization is another way of normalizing the data (used a lot in machine learning), where the mean is subtracted to the image and divided by its standard deviation. It is specially useful if you are going to use the image as an input for some machine learning algorithm, as many of them perform better as they assume features to have a gaussian form with mean=0 and std = 1.

Data stretching:

Histogram stretching when you work with images.

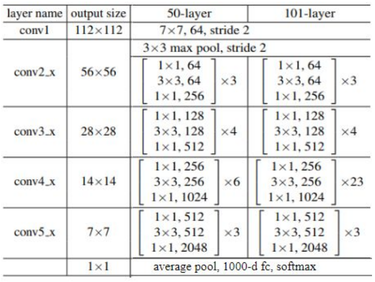
We chose data standardization in our model. Image input is standardized by subtracting mean from each input pixel and then dividing the result by the standard deviation. Image normalization will eliminate the influence of high frequency noise. More importantly, it would make convergence faster while training the network.

4. Experiments and Analysis

Experimens consists of five stages: model selection, model testing, model training, analyze overfitness.

4.1 Model Selection

We followed several principles during model selection. When deeper networks are able to start converging, a degradation problem has been exposed: with the network



layer name	output size	50-layer		101-layer	
conv1	112×112	7×7, 64, stride 2			
conv2_x	56×56	3×3 max pool, stride 2			
		[1×1, 64 3×3, 64 1×1, 256]	×3	[1×1, 64 3×3, 64 1×1, 256]	×3
conv3_x	28×28	[1×1, 128 3×3, 128 1×1, 512]	×4	[1×1, 128 3×3, 128 1×1, 512]	×4
conv4_x	14×14	[1×1, 256 3×3, 256 1×1, 1024]	×6	[1×1, 256 3×3, 256 1×1, 1024]	×23
conv5_x	7×7	[1×1, 512 3×3, 512 1×1, 2048]	×3	[1×1, 512 3×3, 512 1×1, 2048]	×3
	1×1	average pool, 1000-d fc, softmax			

Figure 5. ResNet-50 and ResNet-101 Architecture.

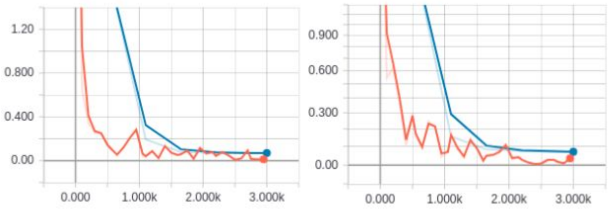


Figure 6. Training (orange) and testing (blue) loss on MNIST with ResNet-50 (left) and ResNet-101 (right)

depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. it is not because of the overfitness of the model and adding adding more layers to a suitably deep model leads to higher training error, as shown in [1]. The dataset we selected as a relatively hard to optimize since the bounding boxes may only contain the partial body of the person and some of the poses do not have enough data to build the ideal model. That's why we choose to use the deep residual learning framework. Each residual unit contains the shortcut output obtained from skipping some previous layers. Thus, during training, the neural network may have chance to skip some layers by learning the output $F(\mathrm{x}) = 0$. For our project, we choose to analyze Resnet-50 and ResNet-101 since they have shown great performances on CIFAR-10 dataset.

4.2 Model Testing

We test our implemented models on the MNIST dataset to verify the performance. Both ResNet-50 and ResNet-101 have good performances, while the 101 layers network have a slightly better accuracy. The following table summarizes the performance:

	Resnet-50	Resnet-101
Training Accuracy	99.70%	99.67%
Testing Accuracy	97.76%	97.98%
Training Loss	0.0121	0.0353
Testing Loss	0.0701	0.0698

Since the result is similar but 50 layers has significantly less training time, we decide to use ResNet-50 on the AVA II dataset.

4.3 Model Training

We first tried the ResNet-101 on our selected subdataset of AVA and the training process is not a success.

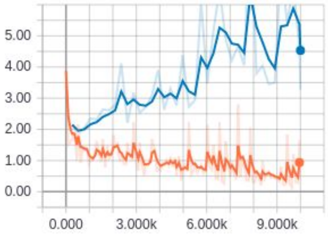


Figure 7. Loss of training data (orange), and testing data (blue) without image augmentation.

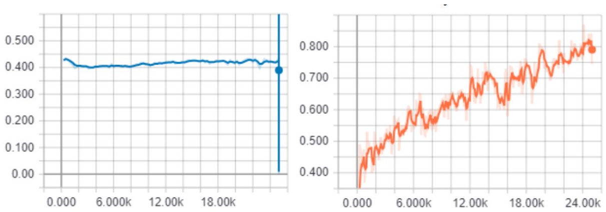


Figure 8. Accuracy of training data (orange), and testing data (blue) with image augmentation

According to Figure 7, the loss of the test data increases as the number of training iteration increases without any improvement at the start of the training process. There are two reasons that leads to this failure: 1. the learning rate is not being set properly 2. the training dataset is not large enough to perform good generalization ability especially when the released dataset has equal size of the training and evaluation data. Thus, we decrease the learning rate and also use a few commonly used data augmentation techniques mentioned in section 3.2. From the experiment result in Figure 9, the loss of test stop increasing as expected. However, the model stopped to learn from step 1000 and started to overfit the training data.

bend/bow (at the waist) (2592):1	'accuracy': 0.112048192	'loss': 2.6152217
crawl (61):2	'accuracy': 0.0	'loss': 5.6407995
crouch/kneel (678):3	'accuracy': 0.0	'loss': 6.268097
dance (1406):4	'accuracy': 0.0	'loss': 8.54392
fall down (114):5	'accuracy': 0.0	'loss': 5.911521
get up (439):6	accuracy': 0.0	'loss': 4.7554007
lie/sleep (1897):8	'accuracy': 0.14117647	'loss': 2.9466526
martial art (624):9	'accuracy': 0.0	'loss': 6.2906637
run/jog (1146):10	'accuracy': 0.0	'loss': 4.610279
sit (30037):11	'accuracy': 0.63776674	'loss': 0.820275
stand (45790):12	'accuracy': 0.21021645	'loss': 2.727881
walk (12765):14	'accuracy': 0.35461974	'loss': 1.58799

The larger the dataset, the higher the performance. Base on the imbalance distribution of dataset. We tried down sampling of data with larger size. But the output accuracy became lower. So we choose to train the model with this imbalance dataset. As you can see, those samples with a higher size of training data would have better performance, including sitting standing and talking.

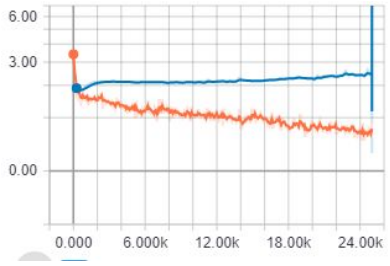


Figure 9. Loss of training data (orange), and testing data (blue) with image augmentation

Moreover, we also realize that the accuracy of sleeping is relatively higher than our data, even if it only have a small data size. We think the reason it happens is that the ResNet is more suitable to this data.

Furthermore, there are some labels with accuracy 0. We think there are 2 main reasons for the 0 accuracy. First, some data with smaller size might not have fully trained compare to those data with larger size. Then it would be hard for the model to capture the features of these labels. The second reason is that there are some labels which is tend to be more dynamic recognizable such as dance. These data would be hard to be recognized by single frames only.

At the end, the result turns out not very satisfiable and we think a sequential model is more suitable to do human action recognition in videos.

4. Conclusion

It's not enough to do activity recognition on video dataset by analyzing single image frames for each activity. The labels extracted, actor pose is the easiest one to predict comparing to person-to-person and person-to-object. With the interaction, there may exists more consequences and dependent actions. Since the video frame prediction on actor pose is not as well as we expected, we end up with a conclusion that using sequential model on analyzing images is necessary.

5. What we learned

1. How to read csv file and convert images to numpy arrays before training. 2. Image augmentation can reduce the overfitness of the model. How to create model presented in a research paper. (not simple CNN, LSTM). 3. How to deal with the problem of ovefitness. How to reduce the memory needed during image preprocessing and training. 4. How to deal with the problem of unbalanced data for each label. 5. Debugging skills such as whether correctly shuffled the data; whether extracted images has assigned a label correctly, why test accuracy doesn't increase, etc. What we could do differently next time is: 1. Double check whether selected actor poses are correctly read from the csv file. 2. Carefully went through each released version of the dataset (1.0, 2.0, 2.1). 3. Shuffle the data before training 4. Save Images into separate batch before training to increase loading speed. 5. Check the training/testing data size for each label before training.

References

[1] He, Kaiming et al. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.

[2] Murray, Naila et al. "AVA: A large-scale database for aesthetic visual analysis." *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012): 2408-2415.