Action Recognition in the Moments-In-Time Dataset

Zihan Lin
ICME
Stanford University
zihanl@stanford.edu

Hao Yin
ICME
Stanford University
yinh@stanford.edu

Jason Zhu
MS&E
Stanford University
jzhu121@stanford.edu

Abstract

We build action recognition models for a newly released dataset of human and animal actions, the Moments-In-Time . We explore and compare two categories of architecture architectures: the purely spatial model and spatio-temporal model. Measured by top-1 and top-5 accuracy, we found that the improvement of spatio-temporal models is insignificant. Looking into the accuracy in details for each classes, we found that for activities that requires temporal information to recognize (such as closing and falling), the spatio-temporal models perform significantly better. Moreover, we found that the CAM (class activation mapping) reveals accurately the time and pixels in the video that is responsible for activity recognition.

1. Introduction

Moments-In-Time [6] is a newly released video dataset. Each video is three-second long, and labelled with the action happening therein. This dataset has high-coverage, high-density, and balanced classes. In total, there are over 900,000 labelled videos from 339 activity classes. A subset of this collection, which contains 200 classes and 500 videos in each class, is used as the dataset for the CVPR Moment in Time Challenge 2018 - Mini Track [1]. This project will be our participation in this challenge, in which we explored several architectures of acitivity recognition models and compared their prediction performance.

We investigated two categories of network architectures. The first category is the purely spatial model where we classify each video based on several single frames of with ingoring the temporal information. This category is our baseline. The second category is the spatio-temporal model where we also take into account of the temporal information. The kernel used in each convolutional layers of this model is a 3-dimensional tensor (2-dimensional spatially and 1-dimensional temporally) which extracts spatial and temporal information. We evaluation the prediction performance of each model by the top-1 and top-5 accuracy on the validation set.

As a side product, we would like our model to be able to localize the features in the video that is responsible of the action recognition. To do this, we generate the CAM (class activation mapping) [11] of each video to highlight the temporal pixels that activates our prediction decision. Due to the restriction of data, we will not test our performance of localization based on some metric, just for fun and demonstration.

We used the ResNet model in the purely spatial architecture, one of the most successful models in image classification [4] . For spatio-temporal models, we only explored the purely convolutional architectures because the videos are relatively short (only 3 seconds long) and this architecture admits feature localization. Specific examples we investigated are the 3D-ResNet and 3D-ResNext [9]. We find that the best model among them is the 3D-ResNext, and 2D-ResNet is the second; however, the performance difference among them is insignificant. Specifically, measuring with the top-1 accuracy, 3D-ResNext is higher than 2D-ResNet by 0.9%, which is then higher than 3D-ResNet by 0.5%.

We speculate that the main reason for this insignificancy is the inadequacy of data volume. Note that we only have 100K videos in total, with 500 in each video class, which is much smaller comparing with the ImageNetdataset for image classification. With this small data classification, the 3D kernels used in spato-temporal models might not get well-trained. Other possible reasons might be our limited computing power and consequently limited choice of model architecture

Besides naively comparing the classification accuracy on the whole validation set, for deeper understanding, we compare the prediction accuracy on each class of activities. We find that for actions that require temporal information to recognize (such as falling and closing), spatial-temporal methods obtains more accuracy; however, for actions that is easily recognized by a single picture (such as

[1] http://challenge.moments.csail.mit.edu/competitions/18

sailing and juggling), 2D-ResNet performs better since we have taken advantaged of parameters that is pretrained on the ImageNet.

Interestingly, despite the relative-low prediction accuracy, our localization experiments show that the network does recognize the important pixels in the video. This finding reveals that our model does capture the important information in each video, and can potentionally predict better with more cautiousness in parameter training and architecture design.

2. Related Work

Training a neural network on video dataset is a hard problem due to the lack of high-quality dataset, high requirement of computing power, and efficient design of network architecture [5]. There are a few other creative methods developed specifically for video classification. The models we explored in this project have many similarities with the two methods mentioned below. For example, Residual Blocks are used in many of the models because of the great performance and the ability to extend to much deeper structure. However, feature localization using the Class Activation Mapping cannot be easily applied with the following two methods, so we just provided a brief introduction.

ConvNet+LSTM [2] The general architecture of this type of network is to apply conventional 2D convolution on each frame of video to exploit the spatial information, and adding upon a recurrent layer on top to utilize their temporal relationship. This network has two main advantages. First, it effectively takes advantage of the state-of-the-art techniques in image classification, such as GoogLeNet and ResNet. Specifically, we can use the low-level layers that have been proved to effectively extract local image features and use it as video spatial features. Second, this type of models allow us to make prediction in an online fashion. Unlike 3D ConvNet, we don't need to look through all the frames in the video; we can make prediction upon any video frame. This advantage also makes the model applicable to video of any length (in time), relieving the effort of video preprocessing. Possible problem within this architecture is that it deals with temporal information in a late fusion manner [5] which is not as effective as slow fusion.

Two-Stream Networks [7] This kind of network architecture provides the current state-of-the-art activity recognition performance on existing medium-size video collection such as HMDB-51 and UCF-101. As implied by the name, the network contains two streams of subnetworks. The first stream is a common 2D convolution on an arbitrary frame of the video, extracting the spatial information. The second stream works on optimal flow of the video clip [10], extracting temporal information through tracking the movement of object in the video.

3. Data

3.1. Number of videos

In this course project, we focus on using the Moments-In-Time Mini track, which is a subset of the original Moments-In-Time dataset. The mini set consists of 200 action classes. In the training set, we have 500 videos for each class, which gives 100,000 videos. In the validation set, we have 50 videos for each class, which sums to 10,000 videos.

3.2. Length of each video

Each video lasts around 3 seconds, and each second contains 30 frames. To apply Convolutional Neural Network to the video data, we have extracted all the frames from the videos, and saved them as jpeg files. In our Mini dataset, the number of frames for each video ranges from 77 to more than 100.

3.3. Spatial Transformation

Each jpeg file has the same image height and width of 240 by 240. We normalize the data by subtracting the mean for each RGB channel from each frame. Because many of the videos have black margins, we crop out only the center 112 by 112 pixels as inputs before feed into our model. For a probability of 0.5, we also apply a horizontal flip to a frame.

3.4. Temporal Transformation

Since each video consists of different number of frames,in order to normalize each input example to the same dimension, we further apply temporal transformation by picking the same number of frames from each video. Due to the limitation on computing power, we only sample 16 frames from the original video and hope those frames contain key information for the model to classify.

4. Technical Approach

4.1. Models

For action recognition, following the categorization in [1], we will explore existing neural network models of three types.

2D ConvNet [8] This is a direct application of the conventional 2D Convolution Neural Network. We sampled 4 equal-distance frames from each video, and apply Resnet to each frame. The ResNet kernels are still 2-dimensional.

The test time classification is just an average over the 4 individual frame level outputs.

To speedup the training process, we used the pretrained ResNet 50 provided by PyTorch, and use transfer learning to only fine-tune the last ResNet block and the fully-connected layer.

3D ConvNet [8] This type of model is a directed generalization of the conventional 2D convolution to the 3D case, where we introduce the temporal dimension in video dataset. It aims to directly capture the spatiotemporal information through the 3D kernels.

Though seemingly intuitive, this method does not perform as well as other two methods [1]. One reason for this inaccuracy lies in the lack of large video dataset (at the magnitude of `ImageNet`) from which good kernals of extracting video features can be obtained. Since the `Moments-In-Time` datasets is a much larger collection of videos, we expect a better performance of the 3D ConvNet method in our explorations.

ResNet has been used to improve image classification performance by adding increased depth to CNNs [4], and the use of very deep CNNs trained on ImageNet have facilitated the acquisition of generic feature representation. [3] extends the residual learning ([4])[ResNet] to a 3 dimensional case, and is the first to consider such deep 3D ConvNets.

3D-ResNext [9] considers repeating a building block that aggregates a set of transformations with the same topology, and exposes a new dimension, which they call cardinality (the size of the set of transformations), as an essential factor in addition to the dimensions of depth and width. [3] adapts the 2D-ResNext to a 3-D setting, and use pretrained parameters on `Kinetics`to obtain state of the art performance on `UCF-101`and `HMDB-51`datasets.

4.2. Pretrain

Recent work [1] has evaluated the performance of the three types of models on the `HMDB-51`, `UCF-101`, and `Kinetics` datasets. Besides comparison, it provides a new model that combines 3D ConvNet with two stream networks. To overcome the difficulty of lack of video samples, this model uses transfer learning to initialize the 3D kernel filters with 2D kernels obtained from `ImageNet` training, which significantly boosts the prediction accuracy.

3D Pretraining: Recent work [3] focus on the training of very deep 3D CNNs from scratch for action recognition. To overcome the difficulty of limited sample size, they use transfer learning with 3D kernel pretrained on `Kinetics` dataset, which is positioned as a de facto video dataset standard that is roughly equivalent to the position held by ImageNet in relation to image datasets. Compared to [1], which use transfer learning with 2D kernels obtained from `ImageNet`training, this paper is the first to consider this 3D Pretraining idea to overcome the prior difficulty in 3-D Convnet training.

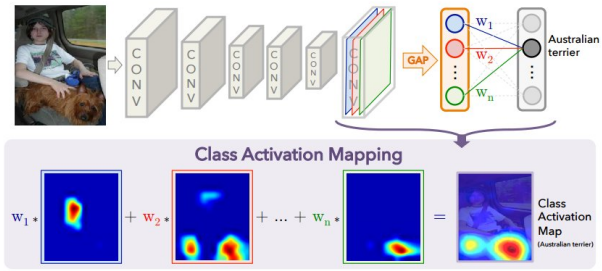


Figure 1: Class Activation Mapping Framework

Method	Top 1 accuracy (%)	Top 5 accuracy(%)
2D-Resnet	18.0	40.7
Resnet-101	17.5	39.8
ResNeXt 101	18.9	41.0

Table 1: Performance Comparison between different models

4.3. Discriminative Feature Localization

For discriminative feature localization, we plan to adopt the Class Activation Mapping method proposed in [11]. All fully connected layers before the output layer are removed from the model since they will mess up with location information of the features. Alternatively, a global average pooling layer is used on each channels of the last Convolutional layer (assume n channels) to produce n neurons. These n neurons are then multiplied by a matrix to produce the logits. The entries of the matrix are then used as weights for calculating a weighted average of the channels of the last Convolutional layer, which is just the heatmap to localize the features. The framework is presented in Figure 1

We use both the top-1 accuracy and top-5 classification accuracy as the scoring metrics. Top-5 accuracy is appropriate for video classification as videos may contain multiple actions within them.

5. Experiments

In this section, we give an overview of the implementation details, present our model performance, and display most confusing classes and compare different models to gain an intuitive understanding of our model and dataset.

Freq	Actual	Predicted
0.460	barbecuing	grilling
0.320	waking	sleeping
0.300	planting	gardening
0.280	emptying	filling
0.260	handwriting	drawing
0.260	boiling	frying
0.200	studying	reading
0.200	folding	crafting
0.200	slicing	chopping
0.200	exercising	stretching
0.200	boating	fishing

Table 2: **Most Confused Categories**: The most commonly confused categories for 2D Resnet

Freq	Actual	Predicted
0.687	barbecuing	grilling
0.448	gardening	planting
0.367	frying	stirring
0.306	sailing	boating
0.285	closing	opening
0.285	barking	howling
0.244	boiling	stirring
0.244	digging	planting
0.229	cooking	stirring
0.224	combusting	burning
0.208	landing	launching

Table 3: **Most Confused Categories**: We show the most commonly confused categories of 3D-ResNext

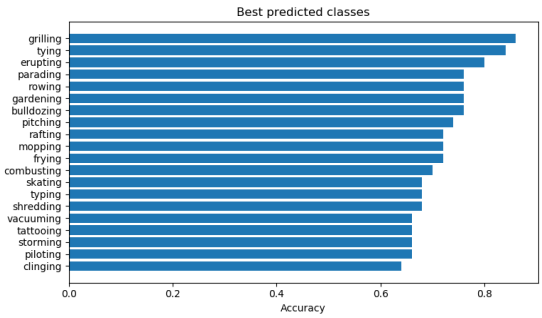


Figure 2: Top 5 accuracy distribution per categories

5.1. Implementation

We use stochastic gradient descent with momentum to train the networks and randomly generate training samples from videos in training data in order to perform data augmentation.

We also perform mean subtraction, which means that we subtract the mean values of `Moments-In-Time` from the sample for each color channel. All generated samples retain the same class labels as their original videos. In our training, we use cross-entropy losses and backpropagate their gradients. The training parameters include a weight decay of 0.001 and 0.9 for momentum.

For 3D ConvNet models, we use the pretrained parameters from [3] on `Kineticsdataset`, and fine tune the parameters of the last two layers (the conv5-x and the fully connected layer). For 2D ConvNet models, we use the pretrained model trained on `ImageNet`, and fine tune the fully connected layer. We also try training without pretrained parameters, but find that pretrained models obtain a much faster convergence speed and a better performance.

For specific model structure and layer specification, We illustrate them in a very intuitive way. See figure 4.[Source: [3]]. One can see these models as a generalization from 2D-ResNet ([4]) and 2D-ResNext ([9]).

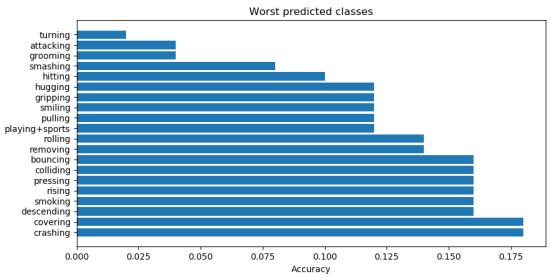


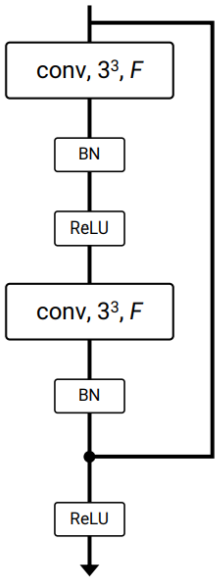
Figure 3: Top 5 accuracy distribution per categories

5.2. Results

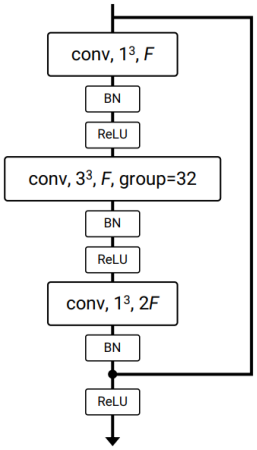
Table 1 summarizes the performance of three major models on `Moments-In-Time` dataset. Although 3D ConvNet models are supposed to capture temporal information and perform better than 2D ConvNet models, this is not the case for our dataset because of the difficulty of `Moments-In-Time` dataset.

This performance is not as good as on other datasets because : 1.We only train the data on a small dataset (due to computational power limitation, we only train on the Moment in Time Challenge 2018-Mini Track.) 2.The underlying difficulty of the task. As can be seen from the next subsection, the model has to be trained to differentiate similar actions such as barbecuing and grilling, gardening and planting, frying and stirring, etc. These tasks can be hard even for human, and thus we do not expect computer vision models to obtain superior performance as in other datasets, such as `UCF-101`, `HMDB-51`.

To understand some of the challenges, Figure 2 and 3



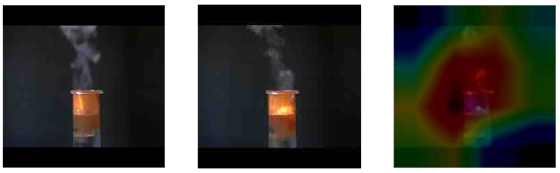
(a) 3D-Resnet Structure



(b) Classification score

Figure 4: 3D-ResNext strucutre

breaks down performance by category for different models and modalities. Categories that perform the best tend to have clear appearances and lower intra-class variation, for example bowling and surfing frequently happen in specific scene categories. The more difficult categories, such as covering, slipping, and plugging, tend to have wide spatiotemporal support as they can happen in most scenes and with most objects. Recognizing actions uncorrelated with scenes and objects seems to pose a challenge for video understanding.



Ground Truth: combusting

2D ResNet Top 5 Predictions:
1: pouring (0.264) 2: combusting (0.226) 3: burning (0.112) 4: launching (0.088) 5: smoking (0.048)

3D ResNet Top 5 Predictions:
1: burning (0.226) 2: combusting (0.185) 3: juggling (0.102) 4: spinning (0.045) 5: pouring (0.023)

Ground Truth: cooking

2D ResNet Top 5 Predictions:
1: grilling (0.620) 2: barbecuing (0.320) 3: baking (0.028) 4: frying (0.018) 5: combusting (0.004)

3D ResNet Top 5 Predictions:
1: grilling (0.664) 2: barbecuing (0.151) 3: baking (0.117) 4: frying (0.033) 5: cooking (0.013)

Ground Truth: handwriting

2D ResNet Top 5 Predictions:
1: drawing (0.728) 2: painting (0.040) 3: licking (0.017) 4: handwriting (0.015) 5: whistling (0.013)

3D ResNet Top 5 Predictions:
1: drawing (0.947) 2: handwriting (0.050) 3: tattooing (0.002) 4: cracking (0.000) 5: knitting (0.000)

Ground Truth: licking

2D ResNet Top 5 Predictions:
1: licking (0.183) 2: yawning (0.136) 3: eating (0.064) 4: sleeping (0.064) 5: resting (0.056)

3D ResNet Top 5 Predictions:
1: yawning (0.116) 2: waking (0.071) 3: crying (0.062) 4: eating (0.054) 5: sleeping (0.053)

Ground Truth: dancing

2D ResNet Top 5 Predictions:
1: cheering (0.304) 2: parading (0.148) 3: dunking (0.055) 4: shouting (0.051) 5: celebrating (0.047)

3D ResNet Top 5 Predictions:
1: protesting (0.575) 2: parading (0.145) 3: shouting (0.040) 4: drumming (0.034) 5: arresting (0.021)

Figure 5: Examples of missed detections

5.3. Most confusing categories

Table 2 shows some of the most common confusions between categories. Generally, the most common failures are due to errors in fine-grained recognition, such as confusing frying versus stirring, barbecuing versus grilling, barking and howling, or lack of temporal reasoning, such as confusing opening versus closing. The confusions between 2D-ResNet and 3D-ResNet are quire similar, suggesting that our 3D Resnet does not resolve the confusions of 2D models.

5.4. Comparison between models

By comparing the accuracy of different models (3D-ResNext and 2D-ResNet), we find that 3D-ResNext outperforms the 2D-ResNet in the following category: 'falling', 0.479 (top 5 accuracy), 'biting', 0.420 (top 5 accuracy), 'filling', 0.380 (top 5 accuracy), 'closing', 0.239 (top 1 accuracy), 'chopping', 0.260 (top 1 accuracy). We find that all of these categories has something to do with actions and temporal information. This confirms our conjecture that 3D Models are better in terms of capturing temporal information and may be do better in identifying actions.

Also, we find that 2D-ResNet outperforms the 3D-ResNext in the following category: 'tattooing', -0.400 (top 1 accuracy), 'juggling', -0.319 (top 1 accuracy), 'sailing', -0.400 (top 5 accuracy). For these categories, you don't really need temporal information for correctly identifying an action, but rather capture the information from a frame of the video. Also, because the number of pixels we consider in the 2D-ResNet is larger than that of 3D-ResNet, it makes sense that 2D-ResNet is able to perform better than 3D-ResNext.

5.5. Feature Localization with CAM

After the training of the classification models, we adopt the method in [11] for feature localization. The output of the last ResNet block and the Average Pooling layer for each image, and the model parameters of the model output weighting matrix have been extracted to calculate the heatmap, which should capture the location of the key features of each channel of the Class Activation Map. From the results we see that this method has accurately localized the key objects in the videos.

6. Conclusion

All three methods explored in our report give reasonable performance on the `Moments-In-Time` Mini dataset. We found that the spatio-temporal methods performs much better in recognizing activities that requires extensive temporal information, such as falling and closing. The great performance of CAM reveals that our model can accurately focus on the important time and pixel in the video, showing the great potential of the model if given larger data volume and computing power.

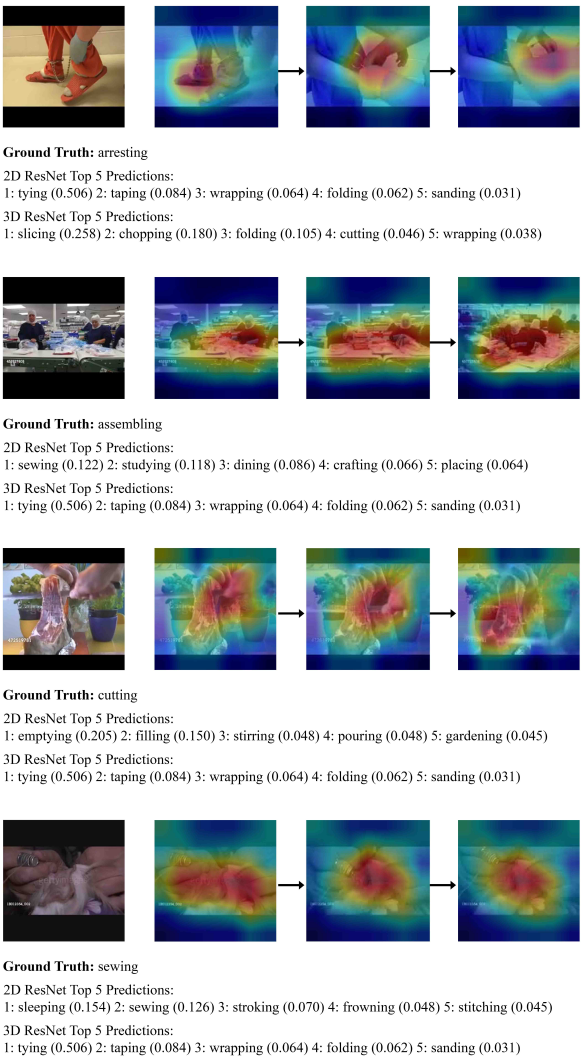


Figure 6: Examples of CAM applied to videos (wrong classification but great localization)

7. Future Work

`Moments-In-Time`dataset presents a difficult task for the field of computer vision in that the labels correspond to different levels of abstraction (a verb like "falling" can apply to many different agents and scenarios, involving objects of different categories). Thus it will serve as a new challenge to develop models that can appropriately scale to the level of complexity and abstract reasoning that a human processes on a daily basis

We believe that the performance of our approach is currently limited by the size of the dataset and the number of

layers of our 3D-ResNet. If we have more computational power in the future, we can preserve more pixels when performing spatial transformations on figures, and explore deeper ResNet on the full `Moments-In-Time` dataset with parameters fine-tuning.

In addition, inspired by the strong performance of CAM, we might consider other tasks that cares less on the classification side of the problem, but focuses more on the feature localization. We believe that CAM can be extended to many other fields to help understanding and interpreting deep Neural Networks.

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[2] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[3] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *arXiv preprint arXiv:1711.09577*, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[6] M. Monfort, B. Zhou, S. A. Bargal, T. Yan, A. Andonian, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding.

[7] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4489–4497. IEEE, 2015.

[9] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[10] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

[11] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.