# Pose Guided Visual Attention for Action Recognition

Ifueko Igbinedion
Stanford University
ifueko@stanford.edu

## Abstract

*When performing visual tasks, it can be argued that humans do not focus on an entire scene at one time. Rather, we focus our attention on important parts of the scene, using contextual cues such as objects and human pose to help guide our attention. This project proposes the use of pose to guide Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units for action prediction. The proposed model is evaluated on the Stanford 40 Actions dataset, producing results that support further exploration of guided attention in RNNs.*

## 1. Introduction

Visual cognition research has noted that humans sequentially focus their attention on different parts of the scene, extracting relevant information for completing tasks [8]. In recent years there has been an increase in interest in attention based models for several visual cognition tasks, including including caption generation [14], visual question answering [13], and action recognition [10]. These models often utilize RNNs with LSTM units, which have shown promising results in learning sequence based problems.

This project aims to tackle to problem of human action recognition in images. While not traditionally a sequence based problem, the idea of visual attention can be modeled as a sequence based problem, in which each area of focus is an item in the sequence of focus areas used to classify the action, and each area influences the next focus area. We propose the use of a recurrent neural network with three LSTM units. The network aims to learn hidden output of each LSTM unit as an attention map for the current input.

In this paper we first discuss related work in the areas of action recognition, attention based networks, and pose estimation. Then we outline the proposed technical solution to the problem, noting the dataset used, feature extraction, and how the network was implemented. Finally, we discuss the results and key insights gained from experimentation.

### 1.1. Related Work

*Action Recognition*

Recognizing human actions in images has many useful applications including image retrieval and scene understanding. There have been many recent network designs proposed to solve the task of action recognition, many of which use RNN variants. Yeung et al. [17] define a novel form of the LSTM for action recognition to model temporal relationships between frames in a video by utilizing multiple input and output connections. Due to this connective modeling, the model is able to refine its predictions in retrospect after seeing more frames. Yao et al. [15] propose a mutual context model to model the co-occurrence of objects and human pose in order to predict interaction. The researchers introduce a set of atomic poses that allow for flexibility in classification and extensibility to additional datasets and activities. Rosenfeld et al. [9] train a series of neural networks to localize and extract hand and face regions for input into a hand-object interaction classifier. They show that recognition is improved by precise localization of the object related to the action, and consequently extracting information of the object together with the human in the interaction. As a result, they achieved an improvement of 35% over the state-of the art. Sharma et al. [10] train soft attention models using an LSTM based recurrent neural network for action prediction in videos. The researchers train a multi-layered RNN with LSTM units to detect which portions of the set of frames are relevant for the current task before predicting the video actions.

*RNNs with Long Short-Term Memory*

Recurrent neural networks, especially those with Long Short-Term Memory, have proven to be very effective at learning models with large amounts of sequential data because they are deep both spatially and temporally. Lately, they have become extremely popular for a wide variety of tasks, including language modeling, machine translation, speech recognition, and image caption generation [18]. Karpathy et al. [6] analyze the strengths and limitations of LSTM based RNNs, finding that while they often show impressive performance, further architectural innovation is needed to eliminate error entirely. Zaremba et al. [18] show how to apply dropout to LSTMs, showing that when used correctly, it substantially reduces overfitting in many tasks. Its wide application set and utilization in many state of the art algorithms suggest that LSTM based RNNs are promising and warrant further exploration.

*Attention Based Networks*

Attention models that utilize RNNs have proven to be useful in a wide variety of applications. Wang et al. [11] give a theoretical and mathematical overview of attention-based recurrent neural networks, showing their flexibility of input and output as well as the different types of models and training schemes. They also define the two main attention mechanisms for RNNs, soft and hard. Xu et al. [14] implement both hard and soft attention mechanisms in networks trained to repeatedly weigh locations of the image based on the importance of the current word being predicted in caption generation. Xu et al. [13] define a multi-hop spatial attention memory network for visual question answering. The researchers implement soft attention models to train a recurrent neural network that generates answers to questions about images. In this network, the question is input as a variable length sequence of words due to the flexibility of the model. Gregor et al. [4] developed an auto-encoder network which utilizes a novel spatial attention mechanism that mimics the natural movement of the human eye in order to draw digits. Bahdanau et al. [2] address the sequence to sequence learning problem for machine translation. Using an attention mechanism, the researchers improve poor performance of direct translation in their network.

*Pose Estimation*

An important component of this project was choosing a suitable pose estimation algorithm, as not every action recognition dataset includes pose annotations. Recently, there have been several promising networks designed to tackle the task of pose estimation. Haque et al. [5] train a convolutional neural network using soft attention models and depth maps to estimate human pose in 3 dimensions. In this network, glimpses of body part information is extracted and used as input into the network. Carreira et al. [3] propose an iterative, self correcting convolutional neural network for human pose estimation that utilizes stochastic gradient descent to infer a set of 2D keypoints describing body pose from a single RGB image. Wei et al. [12] utilize multi-stage pose prediction framework incorporating convolutional neural networks for feature extraction. Newell et. al. [7] introduce the stacked hourglass network, a framework for pose estimation that samples and combines features from high to low resolutions in order to predict joint probabilities at each pixel. By modeling human pose structure via feature extraction, the network allows for higher precision in estimation of joints, even when occluded, and achieves the current state-of-the-art percentage of correct keypoints. This project uses the stacked hourglass network for pose estimation.

## 2. Technical Details

### 2.1. Problem Statement

This project aims to solve the problem of single action recognition in images using pose information. Given an image, we want to first estimate the joint locations of the center person and then predict what action is occurring out of a set of actions. To solve this problem, we propose an LSTM based recurrent neural network that takes in either an image's features or the

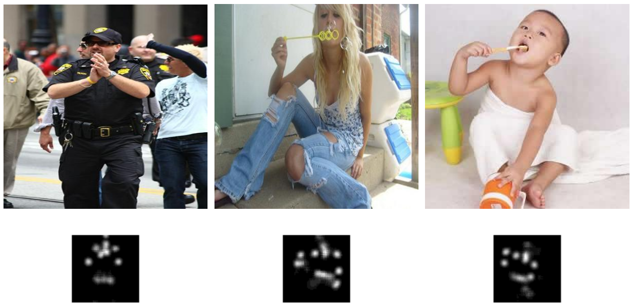

Figure 1. Example Images from the Stanford 40 actions dataset and their corresponding heat maps.

features and the pose map and outputs the predicted action.

*Network Inputs*

We explore two types of inputs. In the first input, we take the image, concatenate the pose information to the bottom of the image and extract GoogLeNet features. From there, we vectorize the image and input this vector into the network. In the second type of input, we take only the images GoogleNet features as the initial input, and use the pose map as the initial hidden state to the first LSTM unit in the network.

*Dataset and Convolutional Features*

The dataset used in this project is the Stanford 40 Actions Dataset [16], which contains 9532 images of humans performing one of 40 Actions. For each input, we use the GoogLeNet model, pre-trained on Imagenet. Taking the last convolutional layer of size [7 x 7 x 1024], we reshape and resize the layer to a single vector of size 4096 to speed up computation time. Pose data is estimated using the stacked hourglass network for pose estimation. We take each of the 16 outputted heatmaps and take the max across all maps, coming up with a heatmap similar to those in Figure 1.

*Long Short-Term Memory units*

In our network, we use the variant of the LSTM unit defined by Karpathy et al. [6]. Figure 2 shows a diagram of the unit. Each contains four gates, input $i_t$, output $o_t$, cell $c_t$ and forget $f_t$, and one output, the hidden state $h_t$. Their states are governed by the equations below:

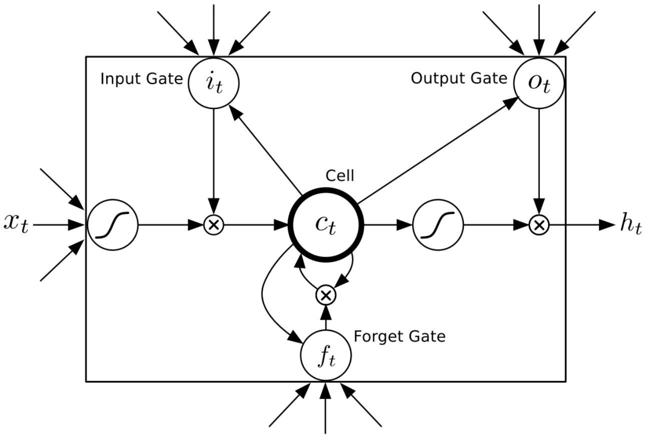

Figure 2. Schematic of a single LSTM unit. Source: http://blog.otoro.net/2015/05/14/long-short-term-memory

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W_l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^{l} \end{pmatrix}$$

$$c_t^l = f_t \odot c_{t-1}^l + i_t \odot g_t$$

$$h_t^l = o_t \odot tanh(c_t^l)$$

The $\sigma$ (sigmoid operator) and $tanh$ (tanh operator) are applied element wise. $W^l$ is a [$4n$ x $2n$] matrix, corresponding to the learned weights of the network. The $i_t$, $o_t$ and $f_t$ are vectors in $\mathbb{R}^n$, and control the local state of the hidden vector. The vector $g_t$ is also in $\mathbb{R}^n$, and ranges between -1 to 1. $g_t$ is used to additively modify the cell memory contents, such that during backpropogation the sum distributes the gradients backwards through time.

### 2.2. Network Design

Figures 3 and 4 show the diagram of the network. The first step is to modify the input images utilizing the output from the stacked hourglass pose estimation network [7]. Then, we input this into the first LSTM, which predicts a hidden state and sends that as input into the second LSTM, which produces another hidden state for input into the third LSTM, which predicts an output. We apply the $tanh$ operator to the output to get the final classification scores. After this, we apply

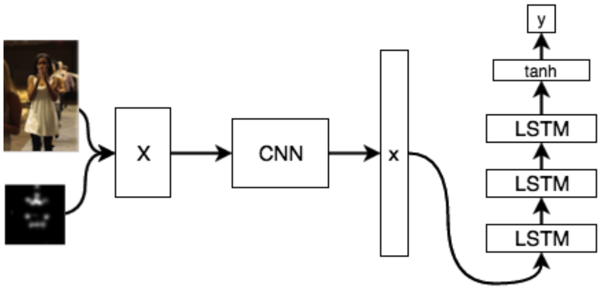

Figure 3. Network with first type of input. Input is the GoogLeNet features of the concatenated image and pose map.

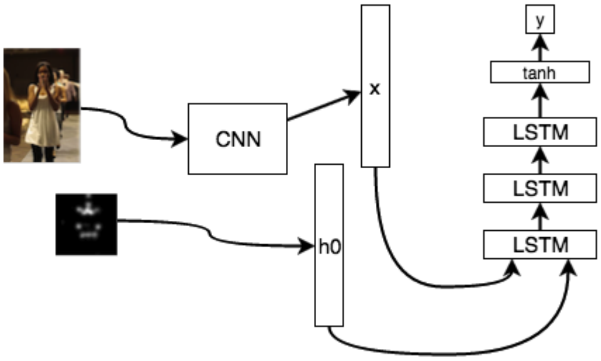

Figure 4. Network with second type of input. Input is GoogLeNet features of image, and initial hidden state is pose map.

batch normalization in an attempt to prevent overfitting. The class with the greatest score is predicted as the output.

## 3. Experimental Setup

The network is implemented on an NVIDIA Tesla K40c GPU using Torch7, with dependencies on CUDA and the following torch libraries: cutorch, cunn, cudnn, cltorch, and dp. We extend the LSTM implementation from [6], using 3 individual units as described above. We use a multi-class Cross Entropy loss function, as defined by the built in class `nn.MultiLabelSoftMarginCriterion`. We evaluate both models on 3 subsets of the Stanford 40 Actions [16] dataset: the complete set, a subset with 25 actions from each class (Small), and a subset with 25 actions from 15 classes (xSmall). We run both networks on all three datasets, calculating classification accuracy on the train, validation and test splits. We run training for 350 epochs of 500 iterations each, with a batch size of 200 examples and a learning rate of 0.001. With pre-computed inputs, each model takes approximately 6 hours to train. Table 1 shows the results.

*Results & Discussion*

*Quantitative Analysis*

For the first input, we achieve high training accuracy on the Complete, Small and xSmall datasets. Validation and test accuracy for the complete dataset is decent, around 0.50 for both cases. The small and xSmall datasets, however, do not perform well in validation and test, the Small dataset achieving 0.24 and 0.08 validation and test accuracy, and the sSmall dataset achieving 0 accuracy for both validation and test. For the second input, we achieve similar results, however, the complete set achieves lower accuracy in all three categories, while the small and xSmall datasets achieve slightly higher validation and test set accuracies.

The first model seems to generalize large amounts of data very well, but vastly overfits the small sized sets. This makes sense: Because the LSTM is spatially deep, fitting a small amount of data is reduced to the task of memorizing that data. As a result, the model overfits the training data heavily. The second model overfits the small datasets as well for the same reason. On the complete set, however, the second model performed more poorly than the first model, which is not what we would expect, but can be explained by the inherent structure of the model. While we provide the initial hidden state to the model and hope that the model learns to predict the next hidden state as an attention map, we cannot guarantee that the model will learn this because we only backpropogate on the output classification labels.

*Qualitative Analysis*

A qualitative evaluation metric utilized was to look at the predicted hidden states of the LSTM for various images. Figure 5 shows the hidden states for the first two LSTM units for the image of the child brushing their teeth in Figure 1. The hidden states do not look like attention maps, but noisy speckles all around the image, which is not what we desired. One way to reconcile this would be to first train the bottom two LSTM units to predict pose embeddings based on the

Table 1. Train, Validation and Test Accuracy for both types of inputs on all three datasets.

| Dataset type | Input type | Train Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|---|
| Complete | Pose concatenated to image | 0.82 | 0.525 | 0.493 |
| Small | Pose concatenated to image | 0.97 | 0.24 | .08 |
| xSmall | Pose concatenated to image | 1 | 0 | 0 |
| Complete | Pose as initial hidden state | 0.796 | 0.36 | 0.328 |
| Small | Pose as initial hidden state | 0.95 | 0.2 | 0.15 |
| xSmall | Pose as initial hidden state | 1 | 0.12 | 0.2 |

Figure 5. Hidden states for first two LSTM units for the third image in Figure 1. The top image corresponds to the output of the first LSTM unit, and the second corresponds to the output of the second unit. These states are not visually informative and can explain poor performance of the second method.

first pose map, and then train the last LSTM unit and the following network layers to predict classification from these pose embeddings.

While the results are decent for large amounts of data, they don't come near the state of the art for a couple of reasons. First, due to time constraints, the model was only trained for 350 epochs of 500 iterations each, as stated above. Because of this, the models were not fully trained, and would likely take days to weeks for the training loss to reach zero. Additionally, using a learning rate of 0.001 may have been too fast, allowing the model to slightly overfit training data in all cases. If time had permitted, a good experiment would have been to run both models with increased iterations and batch sizes, but with a decreased learning rate, and noting whether or not accuracy increased.

## 4. Conclusions

This project has shown that pose may be useful in guiding attention for action recognition. While these models only achieved over 50% and 30% testing accuracy, they have shown that the LSTM architecture is flexible enough to learn many types of inputs, and that human pose can provide good contextual features to guide action recognition. Initially, we predicted that the second model, because of its more complicated structure, would achieve better results when compared to the first model. This was not the case, and led to the insight that the direct output of pose works better as a feature vector than an initial hidden state. Because these models utilize the output of another model, rather than ground truth annotations for pose, The model may be learning incorrect parameters due to inaccurate pose estimation. Additionally, resizing the GoogLeNet features may result in inaccuracy of features, making training a more difficult task.

*Further Work*

There are many future directions this project could take. Because of time constraints, there was not much experimentation done with choosing the learning parameters of the model. One direction is to simply play with the learning parameters until a maximally accurate model is developed, or even to dynamically increase or decrease the parameters based on validation loss.

One proposed input that was not implemented was the creation of an embedding space that transforms each of the joint heat maps into an attention map, to be used as the initial hidden state for the LSTM units. Additionally, the attention mechanism used in this project was to simply use pose as an input or a hidden state, rather than multiply that attention by the input. Another direction could be to implement the LSTM unit that calculates a weight map and multiplies that weight by the input at each step.

This project focused heavily on LSTM units. Another unit we could implement would be the Gated Recurrence Unit (GRU), which was also explored by Karpathy et al. [6], who showed that in many cases the GRU achieves minimal loss as compared to a vanilla RNN or an LSTM. Finally, one major issue is the size of the dataset we used. Because the Stanford 40 actions dataset contains only 9532 images, overfitting may always be an issue. Instead, we could evaluate the models on the MPII human pose dataset [1]. Because this dataset contains over 22k images with pose annotations for each image, it could alleviate the problems of overfitting and inaccurate pose estimation.

## 5. Code Submission

The code for this project was sent as a Google Drive link in the form provided by Kenji on piazza.

## References

[1] Andriluka, Mykhaylo, et al. ”2d human pose estimation: New benchmark and state of the art analysis.” Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

[2] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. ”Neural machine translation by jointly learning to align and translate.” arXiv preprint arXiv:1409.0473 (2014).

[3] Carreira, Joao, et al. ”Human Pose Estimation with Iterative Error Feedback.” arXiv preprint arXiv:1507.06550 (2015).

[4] Gregor, Karol, et al. ”DRAW: A recurrent neural network for image generation.” arXiv preprint arXiv:1502.04623 (2015).

[5] Haque, Albert, et al. ”Viewpoint Invariant 3D Human Pose Estimation with Recurrent Error Feedback.” arXiv preprint arXiv:1603.07076 (2016).

[6] Karpathy, Andrej, Justin Johnson, and Fei-Fei Li. ”Visualizing and understanding recurrent networks.” arXiv preprint arXiv:1506.02078 (2015).

[7] Newell, Alejandro, Kaiyu Yang, and Jia Deng. ”Stacked Hourglass Networks for Human Pose Estimation.” arXiv preprint arXiv:1603.06937 (2016).

[8] Rensink, R. A. The dynamic representation of scenes. Visual Cognition, 7(1-3):17?42, 2000.

[9] Rosenfeld, Amir, and Shimon Ullman. ”Hand-Object Interaction and Precise Localization in Transitive Action Recognition.”

[10] Sharma, Shikhar, Ryan Kiros, and Ruslan Salakhutdinov. ”Action Recognition using Visual Attention.” arXiv preprint arXiv:1511.04119 (2015).

[11] Wang, Feng, and David MJ Tax. ”Survey on the attention based RNN model and its applications in computer vision.” arXiv preprint arXiv:1601.06823 (2016).

[12] Wei, Shih-En, et al. ”Convolutional Pose Machines.” arXiv preprint arXiv:1602.00134 (2016).

[13] Xu, Huijuan, and Kate Saenko. ”Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering.” arXiv preprint arXiv:1511.05234 (2015).

[14] Xu, Kelvin, et al. ”Show, attend and tell: Neural image caption generation with visual attention.” arXiv preprint arXiv:1502.03044 (2015).

[15] Yao, Bangpeng, and Li Fei-Fei. ”Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.9 (2012): 1691-1703.

[16] Yao, Bangpeng, et al. ”Human action recognition by learning bases of action attributes and parts.” Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[17] Yeung, Serena, et al. ”Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos.” arXiv preprint arXiv:1507.05738 (2015).

[18] Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. ”Recurrent neural network regularization.” arXiv preprint arXiv:1409.2329 (2014).