CS 231n: ASL Video Processing and Translation

Scott, Paxton
paxtonsc@stanford.edu

Tlemcani, Rita
ritabt@stanford.edu

Abstract

Despite the large number of American Sign Language (ASL) speakers world wide, until recently, a lack of large data sets resulted in minimal research focus on machine learning algorithms for ASL translation. In the last few years the creation of larger ASL data sets resulted in an increase of papers on both character and word level ASL translation. Our goal with our CS231N project is two fold. First, we will create a basic ASL translation application that provides ASL users with the ability to convert signs into English using a pretrained Two-Stream Inflated 3D ConvNet (I3D) model. Second, we will develop our own ASL world level interpretation model using transfer learning on 3DResNet model [15] described in "SlowFast Networks for Video Recognition" [4] as a primarily educational exercise and compare the results to the I3D benchmark. In this paper, we discuss the current state of the ASL translation field, describe the data set we train our 3DResNet model on, explore both the pretrained I3D model and our 3DResNet model through data visualizations, and present our simple ASL translator web app.

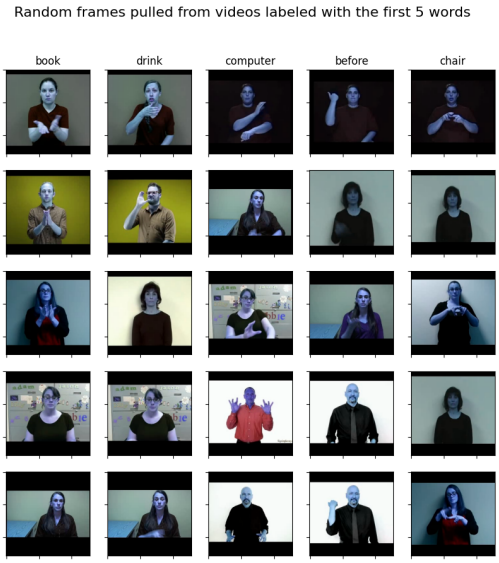

Figure 1. Random Samples from the WLASL[12] dataset

1. Introduction

There are somewhere between 250,000 and 500,00 ASL speakers worldwide. ASL is a visual language that consists of hand gestures, head movement, expression, and body language. In ASL, many of the signs are iconic. For instance, the sign for "eat" is the movement of the fingers and thumb from the dominate hand to the lips. ASL also has an alphabet and speakers can finger-spell words when icon-sign approach is not feasible such as when signing names[9].

Many researchers have tried to solve sign language recognition with approaches that use either RGB-D (D is for an additional depth channel that is produced by devices like Microsoft Kinect) data or raw RGB data. Additionally, researchers have worked both on character-level and word-level translation. One of the historical challenges with ASL translation is the sparsity of data; however recent work has introduced new larger data sets that, combined with transfer learning, produce reasonably high accuracy for word level recognition. For instance, a group[12] from The Australian National University achieved 76% accuracy on labeling signs in 1000 classes using the "top-5" accuracy metric (if the correct word is in the top five classes the sample is marked as classified correctly). Dongxu et al.[12] created their own data-set of 21,083 videos and 119 signers. The researchers then applied four different models to the ASL translation task.

The model which preformed best was the I3D model which is, as described by Dongxu et al., "based on a 2DConvNet inflation: filters and pooling kernels of very deep image classification ConvNets are expanded into 3D, making it possible to learn seamless spatio-temporal feature extractors from video while leveraging successful ImageNet architecture designs and even their parameter".

The I3D model often builds on top of pretrained image classifiers, and then inflates their filters and pooling ker-

nels into a third dimension to allow for video classification tasks. The I3D model builds off previous work on deep image classification networks including Inception[3], VGG-16[14], and ResNet[7]. The I3D model presented in the paper[2] was our benchmark and also the model we used in our web translation application.

The model we developed and trained was created by the Facebook Research team as part of a Project titled "Slow-Fast Networks for Video Recognition" [4] and was based on the 3DResNet[7] model. We applied transfer learning to a model that had been pretrained on the Kinetics 400 dataset[11]. We train the last 5 layers of the 3DResNet network on the WLASL100 dataset[12], which is a dataset of the 100 most common ASL words.

For our model, the input is a RGB video of variable length and size. In our prepossessing algorithm, we reduce the spatial size of the video to 256x256 and reduce the spatial dimension to 64 frames. We then use the 3DResNet model trained on WLASL100[12] to output predictions of the 100 most common ASL words. The dataset we train on includes data on the top 2000 ASL words, but for proof of concept and to make our model trainable we focused on just the top 100.

2. Related Work

Dongxu et al.[12] used different machine learning models based around word-level recognition of RGB video input to interpret ASL. Because most existing datasets were too small and specialized, Dongxu et al. made their own by scraping online videos to create a dataset of 21,083 videos of 119 signers. The videos are labeled with both temporal and spatial bonding. The authors test four models. The first takes RGB data as input, applies a CNN to extract spatial features then feeds the result into an RNN to capture temporal dependencies. The second model also relies directly on RGB data but uses a 3D convolution that extracts both spatial and temporal dependencies simultaneously with a filter that spans both an area of pixels and multiple frames. The third model first extracts the pose data and then passes that data (55 body and hand points in R2) into a RNN. The fourth and last model once again starts with the pose data and uses a temporal graph neural network to extra the human body dependencies before passing each output into the next timesteps input. See Figure 1 for a representation of the four models. The authors also augmented their data through various linear transformations. When judging their classification accuracy, they used only the top K classification accuracy of all the sign instances where k = 1, 5, 10.[12]

Garcia et al.[5] created a real time interpreter that can obtain user video, classify each frame of the video, and then display the most likely word from each source.

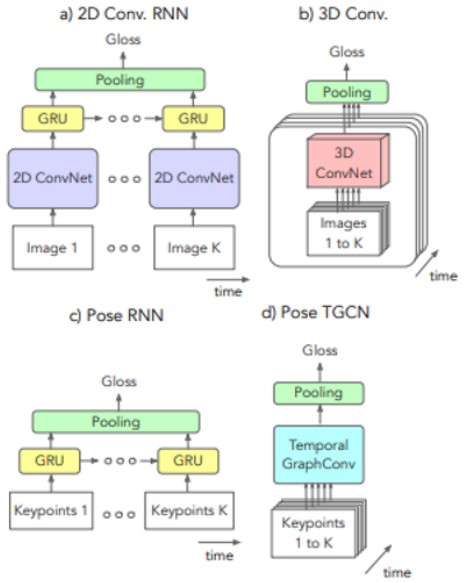

Figure 2. The four models introduced by WLASL paper[12].

They point out that many past applications of machine vision to ASL translation used 3D data. Rather than use an RNN they classify each frame on its own and then predict the likelihood of a string of characters based on their 'language model'. The authors used Caffe to test and run their Convolutional Network. They trained their model on roughly 2500 images cropped around hands. For pre-processing the authors resized the images to 256x256 and then took random crops of 224x224 to further augment the data set. They did center the data, but did not divide by the standard deviation because the RGB data was already in the finite range of 0 to 255. They also mixed in horizontal flips to augment the data set with left/right hand equivalents of each frame. For measurements they used the top-5 accuracy.

Vaezi Joze et al.[10] propose a large data set of 25,000 annotated videos, surpassing the size of data sets currently available for the ASL learning task. The videos in the dataset were taken under unconstrained real-life recording conditions with the help of more than 200 signers which provides a great variation in background and lighting. The dataset covers a large class count of 1000 signs and includes data points from various dialects. The diversity and variation in the dataset means that it can be used to train models that perform well in realistic conditions. The labels for the videos were obtained from subtitles and captions. The proposed method uses the I3D architecture on the videos for ASL recognition. They used the bounding boxes method in addition to face recognition to track the signer. To make sure that the training is signer independent, they divided the

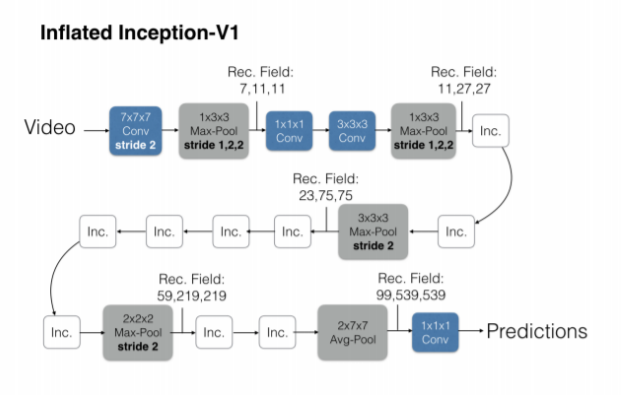

Figure 3. I3d[2] Model Benchmark model used by our web app translator.

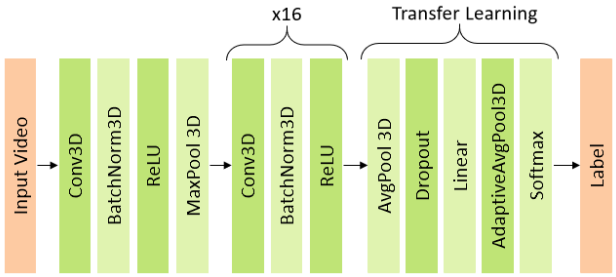

Figure 4. Architecture Representation of 3DResNet[4]

dataset into 80%, 10%, and 10% for train, validation, and test respectively, such that the videos in each set are from different signers. They made 4 subsets available, where each one includes the n most frequent words for n = 100, 200, 500, 1000. They chose the top-5 accuracy as a metric due to context related ambiguity that is characteristic of languages.

3. Methods

3.1. Benchmark Pretrained I3D Method

For our benchmark accuracy we used a pretrained I3D model on the WLASL2000 dataset[12]. Dongxu et al compared different methods in their papers (See Figure 2 for the models they tested) and the I3D model preformed the best. The I3D model was originally introduced in "quo Vadis, Action Recognition? A New Model and the Kinetics Dataset" [2] published in 2018. The ASL I3D model uses pre-trained weights from the original I3D paper. See Figure 3 for a visualization of the I3D architecture.

The I3D model inflates trained 2D ConvNets to handle the temporal dimension of video. The model bootstraps the 3D filters from 2D filters. One bootstrap strategy is to create "boring" videos by simply duplicating a single images multiple times into a sequence. Then the 3D filters can be implicitly trained on a 2D images dataset set such as ImageNet. The expected output for the newly created 3DConvNet is the image class for the single image which can be achieved by making each slice of the 3D filter equal to the 2D filter divided by the length of the 3D filter. Now, the 3D network would provide the same classification for a video created by the duplication of stills as the original 2D ConvNet would have produced for the single 2D image.

The I3D model has four max-pooling layers with stride of 2 and a final average pooling layer. In addition, each of the inception[3] modules have a max-pooling layer. When experimenting with the model, the I3D authors found that it was helpful to not perform max pooling along the temporal dimension in the first two layers. From a theory standpoint, depending on frame rate and the speed of the action, pooling temporal layers can result in conflating separate objects that are only separated by time and not space.

After each convolution layer, the I3D model has a batch-normalization layer and a ReLU activation function. When training the original model, the researchers used stochastic gradient decent (SGD) with momentum equal to 0.9.

3.2. 3D ResNet

The 3DResNet model we implemented (slow_r50) was developed by Facebook AI Research (FAIR) as part of a project called "SlowFast Networks for Video Recognition" [4]. The general ResNet framework originated from a paper released in 2018 titled "A Closer Look at Spatiotemporal Convolutions for Action Recognition" [7].

The SlowFast Network has two pathways for the input video. There is a Slow pathway that only captures a few frames along the temporal axis by using a temporal stride τ. The model trained on the Kinetics 400 used $\tau = 16$ for the slow pathway so 2 frames sampled per second for a 30-fps video. The Slow pathway captures the spatial semantics of the video. The Fast pathway uses a higher temporal resolution with $\tau = 2$. To make the Fast pathway lightweight, it uses a lower channel capacity. The Fast pathway is meant to capture motion information from the fine temporal resolution.

The 3DResNet model is pretrained on the Kinetics 400 dataset[11]. The model consists of 18 blocks where each block consists of 3 to 5 layers. The first Block is the ResNetBasicStem which consists of a 3D convolution layer followed by a 3D batch normalization, a ReLu and a 3D MaxPool layer. See Figure 4 for a representation of the layers used in 3D ResNet. The next 16 blocks are ResBlocks which are each made out of a 3D convolution layer, a 3D Batch normalization, and a ReLU layer. The last block is the ResNetBasicHead which consists of a 3D average pooling layer, a dropout layer, a Linear layer, a 3D Adaptive

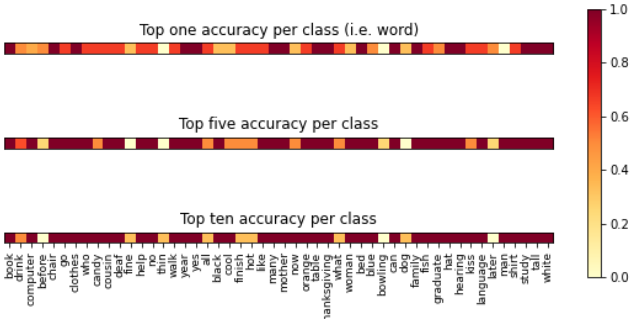

Figure 5. Accuracy of the baseline I3D model word by word on the top 50 words in the WLASL100 dataset.

Average Pooling layer, and Softmax at the end.

We applied transfer learning to slow_r50 by using the pretrained weights on the first 17 blocks and training the last block (the ResNetBasicHead) on the WLASL100. For training, we used a learning rate of 10^{-2} and a weight decay of 10^{-4}. We used the cross entropy function for the loss and we used the recommended[4] optimizer which is SGD with momentum 0.9.

4. Dataset and Features

The dataset we use was created by Researchers from The Australian National University [12]. There are a total of 21,043 video samples with each video ranging from 0.36 to 8.12 seconds in length with an average length of 2.41 seconds. The glosses were gathered from 20 different websites of which many were educational sign language websites. See Figure 1 for a few randomly sampled frames from the WLASL dataset for the top 5 most common words in the dataset. There are four subsets of the whole dataset: WLASL 100, WLASL 300, WLASL1000, and WLASL2000. Each dataset selects the videos with the top-K glosses. In other words, WLASL100 contains all the videos labeled one of the 100 most frequent words. In the WLASL2000 dataset, each gloss has on average 10.5 samples which is quite large by ASL dataset standards. For the WLASL100 dataset which we ultimately used for both training the ResNet 3D model and testing the I3D model, there were 20.3 samples per gloss.

All of the videos contain solely RGB data. When processing the raw video data to test the I3D model, we scaled[1] and cropped the inputs to achieve a frame shape of (3, 224, 224)[6]. Additionally we sample 64 frames at random to enforce a consistent temporal length. To train the 3DResnet, the videos are padded and center cropped to have a unified frame shape of (3, 256, 256). We also sample 64 frames for this model.

The dataset also included a variety of annotations which our model mostly did not make use of. For instance, in the meta data of each video the researchers included body bounding boxes, signer diversity scores, and dialectic variation annotations. The dialectic annotations were hand labeled and used to avoid separate hand signals for the same word in the dataset. We used the signer diversity score to guarantee that the signers in the validation and training data were distinct people to get a validation accuracy that is representative of the real world performance of our model.

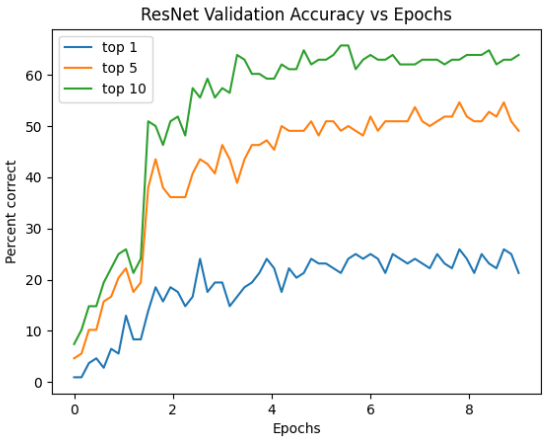

Figure 6. 3D Resnet Validation Set Accuracy

Our split between train, validation, and test set was 80%, 10%, 10% respectively.

5. Experiments

5.1. I3D Baseline

We tested the I3D model pretrained on the full dataset (2000 most frequent words in ASL) and a subset of the dataset of the 100 most frequent words. As an accuracy metric, we used the top-k accuracy method because many ASL words are signed in a physically similar way and are primarily differentiated by context.

On the WLAS2000 dataset, we got 64.32%, 55.81%, and 31.99% for the top-10, top-5, and top-1 accuracy respectively. While On the WLASL100 dataset, we got 90.31%, 84.10%, and 65.89% for the top-10, top-5, and top-1 accuracy respectively. See Figure 5 for a break down of the accuracy that we got on the WLASL100 by word.

To gain a better graphical understanding of the I3D model, we created a matrix of the original video at random time steps and filter numbers for the first three convolutions. The first observation is that each filter decreases in resolution. Right before the last linear layer the visualizations of the filter have lost all human visual significance. However, in the first three convolution layers (See Figures 7 and 8) the original image is not challenging to pick up and we can observe the some filters focus on hands, background,

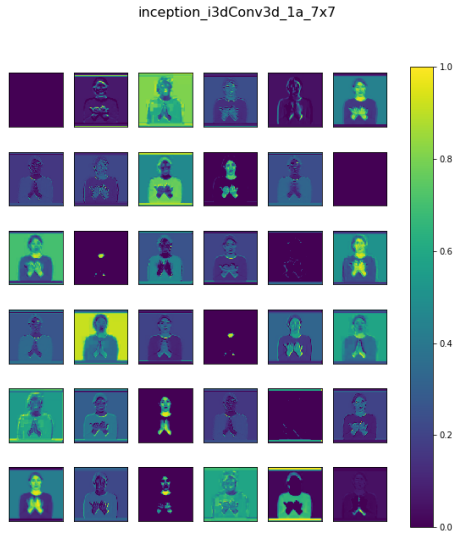

Figure 7. The values of the tensors[13] after being passed through the first 3D convolution in the I3D model with filters of size 7x7x7.

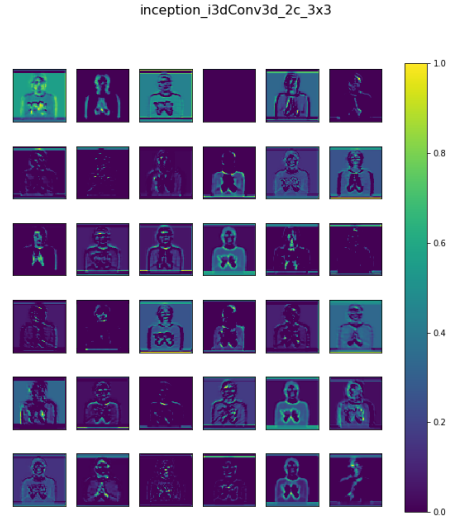

Figure 8. The tensors[13] after passing through the third convolution in the I3D model with a 1by1by1 filter size. We can see that the model is starting to pick up on edge detection with an emphasis on hands.

face, or body. Furthermore, we observe significant edge detection occurring in the second and third convolution layer. Furthermore, by the third convolution layer we can observe that the the majority of the filters are no longer focused on the the background as shown by how much darker the background is in Figure 8 than Figure 7.

To gain a better understanding of the failure points of the model, we graphed the accuracy per word. In the top 50 words, the three worse performers are ”before”, ”later”, and ”bowling”. A commonality with all these words are that the sign primarily originates with wrist gyration and finger movement rather than arm movement.

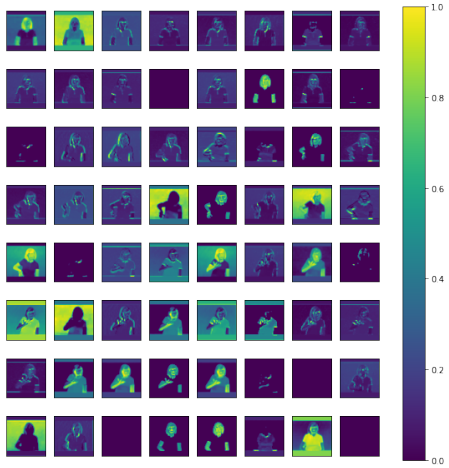

Figure 9. After a sample image is passed through the first layer, titled the ResNetBasicStem layer, some of the output channels focus on background while others focus on the body and face.

5.2. ResNet 3D

Unlike our benchmark I3D model, the slow_r50 model was trained on non-ASL data. We trained our first iteration of the slow_r50 for eight epochs before the accuracy started to plateau. After eight epochs, we achieved 21%, 49.1%, and 63.9% for top-1, top-5, and top-10 accuracy respectively on the WLASL100 dataset. See Figure 6 for a visualization[8] of the accuracy during training.

Similar to the I3D model, when we ran a test video through the nine layers of the model (where each layer in turn is composed of multiple convolutions, normalization, and pools) we noticed that the first layer, see Figure 9, has relatively broad focus with some filters emphasizing the face while other emphasize the hands or the background. After the trial video is passed through the second layer, see Figure 10, the model starts picking up on fine detail with an emphasis on edges.

In addition to passing a test image through the first two layers of the nine layer 3DResNet model, we also graphed two of the 3D convolution filters in the first layer. see Figures 11 and 12. They appear similar to the filters of general image recognition models where the initial filters are larger to learn broad detail and subsequent filters learn finer detail.

6. Conclusion

Over the course of this report, we evaluated the pretrained I3D model, used transfer learning to train our own 3DResNet model, and implemented a simple web interface to act as an ASL translator. See Figure 13 for a screenshot of our current UI along with the top five predicted words.

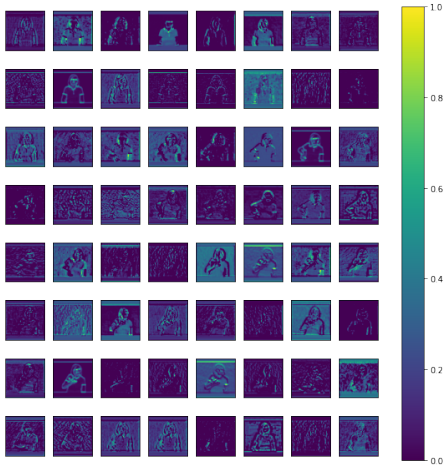

Figure 10. After the test image passes through two layers (each of which consist of multiple convolutions) it is apparent that the model begins to pick up finer details.

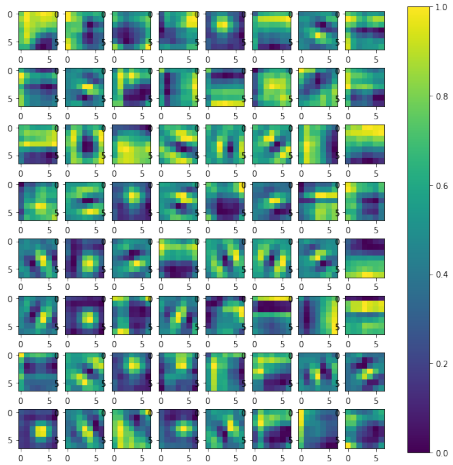

Figure 11. Pictured are the filters in the the first 3D convolutional layer

7. Future Work

The 3DResNet model that we trained did not come practically close to the I3D benchmark. The top-5 accuracy for the 3DResNet model was 49.1% as compared to 84.1% for the benchmark I3D. The next step would be to experiment with our 3DResNet model to achieve higher accuracy closer to the Benchmark. In addition to improving word-classification, it would be interesting to experiment with continuous ASL translation. Possibly the task could be divided into identifying the start and end of distinct signs and then identifying those signs.

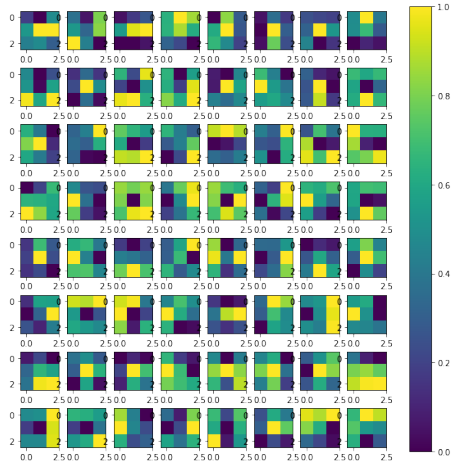

Figure 12. Pictured here are a random slices of the 3x3x3 convolution filters present in the first layer.

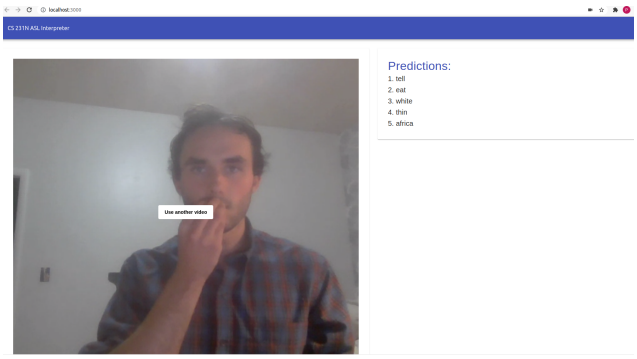

Figure 13. Our simple web app records a user video and outputs the top five word predictions by the I3D model. In this frame, the user is attempting to sign the word 'eat'

8. Appendix A: Web Application

One of our goals going into the project, was not only to experiment with ASl models but also to create a usable ASL translation web application. To do this, we developed a simple front-end using React where the user can create a video of themselves signing a single word. Then (after a pause of about three seconds while our Flask backend considers its options) the top five most likely words pop up. The Flask back-end is currently run on Google Colab and receives the video as a chunk of data in a POST request. The back-end processes the video into a spatial dimension of 224x224 and a temporal dimensions of 64 frames before applying the model to the video and returning the top five results.

This Web App serves as a viable proof of concept for something more complex like a live ASL translator that could take in a continuous stream of signed inputs.

9. Contributions & Acknowledgments

Rita Tlemcani: Reviewed the current literature, provided inspiration for the project, preprocessed the data to interface with the models, set up the Google Cloud machine, transferred and trained the 3DResNet model.

Paxton Scott: Reviewed the current literature, setup the pretrained I3D model, created graphics for the I3D model, created graphics for the ResNet model, created the Web App interface using the I3D model.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018. 2, 3

[3] Yangqing Jia Pierre Sermanet Scott reed Dragomir Anguelov Dumitru Erhan Vincent Vanhoucke Christian Szegedy, Wei Liu and Andrew Rabinovich. Going deeper with convolutions, 2014. 2, 3

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition, 2019. 1, 2, 3, 4

[5] Brandon Garcia and Sigberto Alarcon Viesca. Real-time american sign language recognition with convolutional neural networks. 2

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 3

[8] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007. 5

[9] Michelle Jay. American sign language: Start asl, Feb 2021. 1

[10] Hamid Reza Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language, 2019. 2

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 2, 3

[12] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. pages 1459–1469, 2020. 1, 2, 3, 4

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. Visual Geometry Group, Department of Engineering Science, University of Oxford. 2

[15] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition, 2018. 1