Complexity Beyond the Trigram: Identifying Sign Languages from Video Using Neural Networks

Pamela Toman
Stanford University
ptoman@stanford.edu

Abstract

Being able to distinguish sign languages on the internet holds substantial potential for machine translation and corpus linguistics work. To that end, this paper releases SLANG-3k, a benchmark three-class dataset for sign language identification on the internet. Using the dataset, it establishes that a shallow convolutional neural network with only a sixth of a second of temporal information can outperform a deep network without temporal information in identifying the language in which a person is signing. Incorporating temporal information, even with a loss of spatial fidelity, increases performance from an F_1 score of 0.53 (deep, spatial-only network) to 0.66 (shallow, minimal-time-containing network). Experiments on six distinct architectures incorporating temporal information suggest that more than five frames may be needed to surpass 0.66 F_1 accuracy. Qualitative analyses at the micro-level identify that the network is paying attention to regions with linguistic structure as well as context clues like backgrounds, and at the macro-level qualitative analyses verify performance by reconstructing sign language families. This paper additionally quantitatively assesses the ability of humans to distinguish sign languages given arbitrary still images and estimates human performance at an F_1 score of 0.42.

1. Introduction

This project classifies video segments of sign language as to their language. Using convolutional neural networks with and without temporal information, evaluations are performed on three demographically similar languages for which extensive data and study is available: American Sign Language (ASL), British Sign Language (BSL), and German Sign Language (DGS).

Language identification for signed languages has been minimally studied linguistically and computationally. However, it is a precursor to automatic collection of large corpora of internet-based signed language data for machine translation and corpus linguistics work. This paper finds that for this task, even a three-layer neural network that includes a small amount of temporal structure as per [1, 11, 15] outperforms robust deep feature extraction without temporal structure. This project also establishes the first quantitative evidence that the author is aware of that quantifies the extent to which humans struggle to distinguish sign languages given arbitrary still frames.

This paper makes the following contributions:

- Establishes that the gain from temporal information in sign language identification more than offsets a loss from limited spatial information and computational power
- Introduces a video classification task for which the presence of temporal information can substantially improve accuracy; this responds to previous work that has found that for many classes of video, the performance gain from introducing temporal information is minimal (*e.g.* [15])
- Produces and releases SLANG-3k, a public benchmark dataset for sign language identification on the internet, and presents baseline scores for future work

2. Problem

In order to identify the language used in video clips of sign language, this project compares the performance of convolutional neural networks (CNNs) on still images with their performance on short sequences of video frames.

2.1. Related work

The author was unable to identify substantial machine learning literature on sign language identification in particular. The author is aware of only two papers that perform automatic sign language identification. Gebre, Wittenburg and Heskes [9] extract features from a curated corpus of video clips of 60 seconds and use a random forest model

Figure 1. Still frames of ASL, BSL, and DGS respectively.

to achieve 78% signer-independent accuracy in distinguishing British Sign Language from Greek Sign Language. Gebre, Crasborn, Wittenburg, Drude and Heskes [10] distinguish between six languages using about a half second of curated data with a k-means-based feature extractor and a sparse auto-encoder (a single convolutional layer with ReLU). They achieve best respective average accuracies of 84% and 75% on signer-independent tests.

It appears that minimal work exists in the areas of neural networks applied to sign languages. All identified existing work focuses on handshapes/fingerspelling [20, 22, 38] or individual manual words in isolation [23, 28]; outside these areas, the author is only aware of an unpublished paper on non-manual markers in ASL [35]. Previous work on neural networks for non-sign language identification is also limited. The most relevant paper found is MacNamara and Cunningham [19], who found in 1998 that trigrams on written texts outperform neural networks.

In the neural network literature, the closest task may be action recognition from video. Karpathy, Toderici, Shetty, Leung, Sukthankar and Fei-Fei [15] compare multiple methods of incorporating temporal information into models and identify that temporally fusing convolutions performs best. Additionally, they note that identification of sports using video performs comparably to identification from still images with the exception of sports in which the motion is a defining factor, such as hacky sack. Because motion is fundamental to making sense of signed languages, we introduce the sign language identification task as an action recognition task for which the additional information provided across time is especially helpful.

Language identification on written language achieves accuracies of about 99% [7], and spoken language identification accuracies range between 79-98% [32, 39]. Because techniques that tend to work well on language identification problems have existed for many years, there is little ongoing work on the original forms of these problems.

2.2. Data

For this project, we introduce and release SLANG-3k, a sign-language-on-the-internet identification corpus. SLANG-3k collects 1000 video clips of 15 seconds each of American Sign Language (ASL), British Sign Language (BSL), and German Sign Language (DGS, *Deutsche Gebärdensprache*). The dataset contains the train, test, and validation sets used in the work in this paper (split of approximately 70-20-10%), making future work directly comparable. Clips from any particular source video are only in one set to reduce the possibility of evaluation contamination. However, there is no guarantee that every clip contains a signer in every frame or even at all; this is in keeping with the envisioned use of this technology at scale for identifying corpora.

ASL, BSL, and GDS were selected because a large number of videos are freely available and their users are demographically similar. All the videos were released under Creative Commons. The majority were collected from YouTube through searches for the language name and words like "deaf" in the commonly used written language.

Most videos are monologues, a few are pairwise dialogues, and even fewer involve three or more signers. Some involve an interpreter signing a spoken conversation in which the speakers may or may not be present on screen. As best possible, the dataset excludes signers who seem non-native, children, vocabulary lists, and videos with a substantial number of frames without any signing. We estimate the number of distinct signers in the dataset to be similar to the number of sources per language, as some sources have multiple signers and other signers repeat. Characteristics of the data are provided in Table 1 .

These three languages are distinct from each other. They allow different handshapes, have different syntaxes (BSL is Object-Subject-Verb whereas DGS is Subject-Object-Verb and ASL is Subject-Verb-Object), spell words using different systems (BSL uses a two-handed fingerspelling system whereas ASL and DGS use slightly different one-handed systems), and have limited lexical overlap (estimated at 34% for ASL and BSL) [18, 21, 33, 36, 37]. DGS has some similarity to French Sign Language [36], which ASL is historically derived from; neither has close similarities to BSL.

3. Technical approach

As a still image baseline, this paper transfers learning from Overfeat's layer *fast* model [31], which has six convolutional and two fully connected layers with nearly 150 million weights, and fine-tune the top layers for sign language classification. The paper compares the baseline with temporal approaches of 3 and 4 layer models that among that maximally have around 12 million weights. Overfeat was selected because of its clean embedding in Torch and from an unrealized desire to experiment with image cropping.

3.1. Data processing

From the middle of each clip, one or five consecutive frames are extracted. Preprocessing extracts a square around each person in a frame using OpenCV's [2] Haar cascades to locate faces. Squares the size of the shortest

	Clips	Sources	Hours Sampled	Themes
ASL	1000	103	7.7	religion, infomercial, linguistics
BSL	1000	34	9.8	government, medical
DGS	1000	48	6.9	religion, banking, conventions

Table 1. Characteristics of the language data in SLANG-3k. Each clip is 15 seconds long. Frame rates vary between 28 and 30 frames per second and more than a dozen frame sizes are included.

spatial dimension are extracted centered on the middle of the face. If no people can be identified, the entire frame is resized. If multiple faces are identified more than 3/4 of a frame away from each other, they are extracted separately.

The frames are resized to 231×231 pixels to feed Overfeat and for two of the temporal variants, to 128×128 for another set of temporal variants, and to 32×32 for a final temporal variant. The temporal network variants additionally include data augmentation via horizontal flipping, and their data is normalized using the training sample to have zero mean and unit variance. Overfeat is allowed to perform its own preprocessing.

Attempts to perform background removal using OpenCV had poor qualitative results and were abandoned.

3.2. Temporal convolutions

This paper follows [1, 11, 15] and others in treating temporal information as a highly auto-correlated input akin to the highly auto-correlated spatial inputs. Specifically, this paper builds a four-dimension-per-example dataset in which any pixel is close to its neighbors in time as well as in space. Each example is of the shape *RGB channels* × *temporal depth* × *spatial width* × *spatial height*. The model learns three-dimensional filters that enable correct classification through convolutions across time as well as space, and produces a three-dimensional activation volume for each filter of the shape *filter temporal depth* × *filter spatial width* × *filter spatial height*. See Figure 2.

This work uses a temporal convolution approach to time rather than a long short-term memory network [14] or other recurrent architecture because the language identification task is one for which the author expected that short-term statistical patterns are sufficient to drive much of classification.

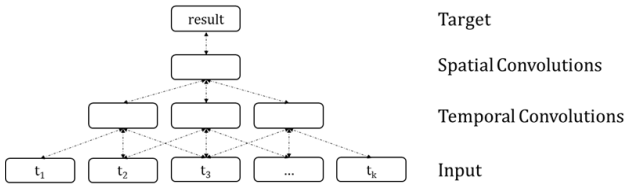

Figure 2. Diagram of temporal convolutions, in which the time dimension has been unwrapped and each type of data appears as a row. Each box has a width, a height, and depth of either channels (at input) or filters (at convolutions).

This is in keeping with the success of even simple models like histograms of trigrams (sequences of three characters) for written languages, as in Cavnar and Trenkle's 99.8% correct classification rate with this method [3]. Although language models often eventually benefit from the additional complexity of cultivating medium- and long-term dependencies [8, 29], the expectation of yet-unwarrantedness for this task was borne out in experimentation that suggests data as the performance limiter.

3.3. Model architectures

At a high level, the primary three architectures are:

- In the single frame model, the first 19 layers of the Overfeat *fast* model [31] transform each image from 231×231 RGB format into a vector of 4096 features. We then continue the training with two convolution-ReLU units and then two fully-connected layers. Each frame is classifed independently.

- In the temporal Overfeat model, the first 19 layers of Overfeat perform feature extraction on a sequence of 5 consecutive frames. We then impose convolutional and fully-connected layers that look at variable depths of the 5×4096 Overfeat outputs to predict the language class. In this setup, initial feature extraction is still limited to interesting spatial properties of each frame independently; however, temporal information is now available to the classifier.

- In the temporal convolutions model, the data are convolved temporally primarily with a depth of 3 to form a visual trigram that allows features of interest to have temporal as well as spatial properties. This architecture learns features from scratch. For increased comparability with the other results, it echoes Overfeat's design (*e.g.*, use of 2x2 pooling, fully connected layers, no batch normalization). However, the author introduced dropout of 50% and L2 regularization to reduce overfitting. A variety of models were trained with this architecture.

See Table 2 for architecture details.

Convolutional neural networks are an unsupervised learning method increasingly used for computer vision. Rather than predefine visual features that are meaningful

Model	Architecture
Overfeat - Still and Temporal	*Convolution (96 filters×11 height×11 width×4 stride) + ReLU + Pool (2×2×2 stride) + Convolution (256×5×5×1) + ReLU + Pool (2×2×2) + Pad (1) + Convolution (512×3×3×1) + ReLU + Pad (1) + Convolution (1024×3×3×1) + ReLU + Pad (1) + Convolution (1024×3×3×1) + ReLU + Pool (2×2×2) + Convolution (3072×6×6×1) + ReLU + Fully Connected (4096 weights) +* Convolution (1024×4096×3 (or 1 for still frame)×1) + ReLU + Convolution (512×1×3 (or 1 for still frame)×1) + ReLU + Fully Connected (256) + Fully Connected (3) + Softmax
Temporal: 231×231×5 (big)	Convolution (96 filters×3 depth×11 height×11 width×1 temporal stride×4 spatial stride) + ReLU + Pool (1 depth×2 height×2 width×1 temporal stride×2 spatial stride) + Dropout Convolution (256×3×5×5) + ReLU + Pool (1×2×2×1×2) + Dropout + Convolution (256×12×12) + ReLU + Fully Connected (3) + Softmax
Temporal: 231×231×5 (small)	Convolution (48×3×11×11×1×4) + ReLU + Pool (2×2) + Dropout + Convolution (96×5×5) + ReLU + Pool (2×2) + Convolution (96×12×12) + Fully Connected (3) + Softmax
Temporal: 128×128×5 (big)	Convolution (96×3×5×5×1×3) + ReLU + Pool (1×2×2×1×2) + Dropout Convolution (256×3×5×5) + ReLU + Dropout + Convolution (256×17×17) + ReLU + Fully Connected (3) + Softmax
Temporal: 128×128×5 (small)	Convolution (48×5×5×5×1×3) + ReLU + Pool (1×2×2×1×2) + Dropout Convolution (96×7×7) + ReLU + Dropout + Convolution (96×15×15) + Fully Connected (3) + Softmax
Temporal: 32×32×5	Convolution (96×3×3×3) + ReLU + Dropout + Convolution (128×3×3×3) + ReLU + Pool (2×2) + Dropout + Convolution (128×3×3) + ReLU + Dropout + Convolution (128×12×12) + ReLU + Fully Connected (3) + Softmax

Table 2. Architectures used. Italics indicate pretrained layers. Pooling refers to max pooling; dropout occurs with 50% probability.

for categorization – which raises substantial challenges regarding what to encode and how best to encode it – the user provides an underlying "network" structure and then feeds (input, correct label) pairs through that structure. Although the network is initialized randomly, it learns from these pairs a set of spatial and/or temporal filters that transform the image data into a form that is increasingly helpful for correctly predicting the target class, such that at the end of training, the user has good filters to use for future inputs for which the correct label is unknown. The network learns these filters through iterative optimization: as each set of pairs passes through the network, the optimization function calculates how much the current filter weight parameters produce incorrect classifications, and then updates the filters' parameters through slight steps in the direction that calculus indicates will improve the final classification. Convolutional neural networks decreased errors in the ImageNet object classification challenge from by over 10% within three years [30], with further improvements since.

Each filter consists of a set of weights on the inputs as they have been transformed so far. During convolution, the filters pass over the current inputs at defined points. At each point, the weights of the filter are multiplied with the corresponding inputs to produce an "activation" for that filter at that point in the input. Thus each layer's output in tandem with filters produces a set of activations to be used as inputs for the following layer's set of filters, until the network designer decides to estimate the target classes or to involve all known activations simultaneously in a non-convolutional fully-connected layer.

As part of the network, it is common to use the rectified linear unit (ReLU) activation function $f(x) = \max(0, x)$ following convolutional layers. This activation function is beneficial in that it increases the nonlinearity of the network, giving the network more representative power; including ReLU layers greatly accelerates convergence [17]. It is also still somewhat common to use max pooling layers, which downsample the current set of activations from each filter to keep only the maximum value in each k-by-k region. Given $k = 2$, $g(x)_{(i,i)} = \max(x_{(i,i)}, x_{(i,i+1)}, x_{(i+1,i)}, x_{(i+1,i+1)})$. Pooling in tandem with downsampling through non-padded convolutions are used in this paper to reduce the data from high dimensionality to three target classes. This paper additionally uses dropout layers, which help regularize and reduce overfitting through zeroing out each feature's activation with a probability of 0.5 during each iteration of training, as per [34]. Dropout encourages redundancy in the network and discourages tightly bound co-evolved features that may overfit the training data.

To estimate the usefulness of the current network weights, this paper uses the cross-entropy loss function with L2 regularization. Under the cross-entropy loss function, we interpret the raw scores for each class that emerge for a particular training example to be unnormalized log probabilities and interpret those values as probabilities using the softmax function. Formally, an example of class j whose raw scores for each possible class i are s_i has a cross-entropy loss L^k on a single example that is defined as:

$$L^k = -\log \frac{\exp s_j}{\sum_i \exp s_i}$$

To reduce overfitting, this paper also discourages weights that deviate far from zero through L2 regularization, resulting in a total loss $\mathcal{L}$ across all N examples of:

$$\mathcal{L} = -\frac{1}{N} \sum_k \left(\log \frac{\exp s_j^k}{\sum_i \exp s_i^k} \right) + 0.5\lambda \|W\|^2$$

where W are the learned weights and λ defines the desired trade-off between high probabilities and weights that deviate from zero.

By optimizing on this function, we perform minimize the negative log likelihood and perform maximum a posteriori (MAP) estimation to find a set of weights that simultaneously maximize the likelihood of the correct class while encouraging weights to fit the prior expectation that they be near zero.

3.4. Training and hyperparameters

Training was performed on a commodity CPU with mini-batches of size 128 for Overfeat-based models and size 30 for temporal convolution models. These batch sizes were selected so that each Overfeat training iteration could occur in 1 second or less, and each temporal convolution training iteration could occur in 45 seconds or less. A CPU was used rather than a GPU because the available GPUs were only able to accept extremely tiny batch sizes, resulting in overall lower performance.

Training used Adam [16] as the update rule to speed learning by adjusting step sizes on a per-parameter basis, with a learning rate selected for each model to minimize the loss and maximize the speed of convergence. The author experimented with a variety of learning rates, comparing training results to validation results on a single held-out validation set consisting of 221 examples (approximately 10% of the total data). Through changing the model structure to include dropout and increasing L2 regularization, the author reduced initial overfitting from 98% training accuracy and 47% validation accuracy to 87% training accuracy and 76% validation accuracy. The author monitored training and manually divided learning rate by 5 whenever loss plateaued for more than two epochs.

4. Results

4.1. Experimental results

This paper reports accuracy and F_1 scores for each experiment. Simple accuracy is the number of correct classifications divided by the number of total classifications:

$$a = \frac{\sum_i f(x_i) = y_i}{N}$$

where $f(x_i)$ is the class assigned to example i and y_i is the truth label for that example.

Because simple accuracy can be misleading especially in cases where the classes are unbalanced, this paper also provides confusion matrices and F_1 accuracy scores. The confusion matrices provide raw results rather than a summary statistic. F_1 scores combine the precision and recall for each category using the harmonic mean. Precision p_i for class i is the fraction of retrieved results that are relevant (the confusion matrix C normalized column-wise for i: $p_i = \frac{C_{(i,i)}}{\sum_j C(j,i)}$). Recall r_i for class i is the fraction of the relevant material that is retrieved (the confusion matrix C normalized row-wise for i: $p_i = \frac{C_{(i,i)}}{\sum_j C(i,j)}$). This paper reports the mean balanced F_1 score across N classes:

$$F_1 = 2 \cdot \frac{1}{N} \sum_i \frac{p_i \cdot r_i}{p_i + r_i}$$

The F_1 score punishes disproportionally low scores on either of the two composite metrics.

4.1.1 Still frame task difficulty: Human

Sign language identification with limited temporal information is challenging. Phonemic contrasts represent the most basic statistically meaningful unit in a language. In sign languages, phonemic contrasts are the result of five features: movement, handshape, location, palm orientation, and facial and body expression (non-manual markers) [37]. Different languages have different tendencies to manifest particular combinations of features. However, given stills from a video, no more than four of these features are clearly apparent, and less information is available the more the frame rate decreases and moving details blur.

To obtain a quantitative estimate of the difficulty of the language identification task for humans, the author developed a survey. Twenty nine respondents identified through snowball sampling were asked about ten images from SLANG-3k; five respondents were used for pretesting the survey instrument. Respondents were asked to rate their level of proficiency in ASL, BSL, DGS (see Table 4), and then to rate each of the images for each language on a scale from *(1) Definitely* to *(5) Impossible*, with an unknown option available. Respondents also were asked to identify when they recognized the signer in the image; those response items were discarded. A screenshot of the survey appears in Figure 3 .

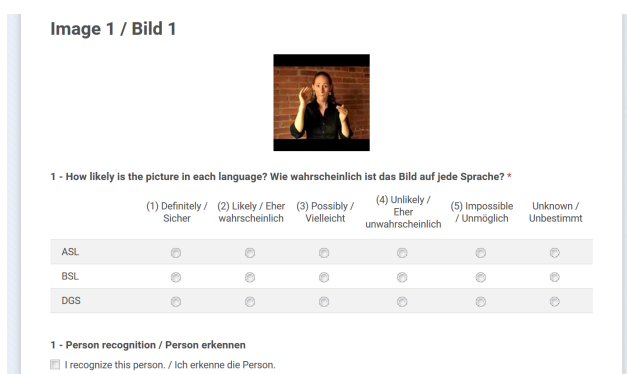

Figure 3. Screenshot of human language survey.

Model	Train	Validation	Test	Indep. Test	Test F_1
Overfeat - Still	81%	61%	55%	98%	0.53
Overfeat - Temporal	97%	69%	68%	3%	0.68
Temporal: 231×231×5 (big)	88%	65%	66%	100%	0.67
Temporal: 231×231×5 (small)	88%	66%	68%	100%	0.66
Temporal: 128×128×5 (big)	89%	76%	65%	100%	0.67
Temporal: 128×128×5 (small)	85%	68%	62%	100%	0.64
Temporal: 32×32×5	80%	72%	61%	97%	0.62

Table 3. Model performance seems to be limited to around 0.66 given only five frames of temporal information. The train/validation/test/independent test columns give simple accuracy performance on each set; the test F_1 score column gives F_1 accuracies on the test set. Scores vary by about ±2% or 0.02 F_1 based on the random weight initializations.

Language experience	ASL	BSL	DGS
Minimal or no exposure	4.2%	75.0%	95.8%
Substantial exposure or fluent	95.8%	25.0%	4.2%

Table 4. Percentage of human respondents at each experience tier.

A human ensemble attains 49.7% accuracy and a 0.42 F_1 score. Accuracy scores and a confusion matrix for the human ensemble are derived by recoding each person's response to each image to be the language that they thought was most likely among their "definitely" or "very likely" responses; ties between multiple language produced split votes. The confusion matrix appears in Table 5.

Written responses from respondents reflected on the difficulty of the task:

> Several of the pictures were blurry or the hand shape/position was unclear, making the task difficult, if not impossible.
>
> Some of the handshapes were possible as signs or "in between" signs for ASL and BSL.
>
> I have used ASL for 23 years. I know that while some signs may be similar to other sign languages, it is very hard to know the sign from a still photograph, even if it is in the language in which you are fluent.

These contextualizing responses indicate that challenge of this task resides in video quality, the arbitrariness of time slice selection, and the inherent linguistic difficulty.

4.1.2 Still frame task difficulty: Model

The single-frame model achieved 54.6% accuracy and a F_1 score of 0.53 (confusion matrix in Table 6). In concert with the human performance of 0.42, these scores substantiate the claim that sign language identification is a challenging action identification task for which still frames are insufficient. Although the model scores are slightly higher than

	ASL	BSL	DGS	Unknown
ASL	51.3	5.3	2.3	22
BSL	28.7	12.7	4.7	23
DGS	19.0	12.5	7.5	32

Table 5. Confusion matrix for human performance (N=24 respondents on 10 images: 24 ASL, 13 BSL, and 2 DGS users; respondents who recognized signers are excluded from those items). Rows are truth; columns are predictions. Accuracy of 49.7%; F_1 score of 0.42.

the human benchmark, the values are not directly comparable. The human respondents were not provided additional training examples, and the model scores may be inflated due to the presence of YouTube series in the dataset that repeat backgrounds, watermarks, or persons across training and testing, as the data do not control for series.

4.1.3 Temporal task difficulty

All models seemed limited to around 66% accuracy on the test set given 5 frames of temporal information. (A representative confusion matrix appears in Table 7, and table 3 provides the performance of every model.) Despite tuning hyperparameters by hand, the author was unable to improve accuracy above about 66%. It is possible that 66% is a bound on accuracy given a sixth of a second of information. Theoretically this is reasonable; a sixth of second is insufficient time to form many complete contextualized signed units, and given the arbitrariness of selected frames, the content might amount to only transitional movements without primary linguistic content at all. The 66% value

	ASL	BSL	DGS
ASL	186	0	19
BSL	99	87	45
DGS	125	1	75

Table 6. Confusion matrix for deep network single frame performance. Rows are truth; columns are predictions. Accuracy of 54.6%; F_1 score of 0.53.

	ASL	BSL	DGS
ASL	132	17	56
BSL	77	153	1
DGS	63	5	133

Table 7. Confusion matrix for multi-frame performance on the 128×128×5 model (big). Rows are truth; columns are predictions. Accuracy of 66%; F_1 score of 0.66.

also echoes the feedback of some human respondents that sometimes it was only possible to rule out one language due to the image structure.

Although smaller spatial sizes of 128×128 pixels compared to 231×231 pixels meant substantially faster training time, that benefit was not similarly realized when images were scaled down to 32×32 pixels – despite the use of a deeper network for the 32×32 task. At that resolution, the loss of detailed spatial information had an adverse effect on performance. As such, this paper recommends scales near 128×128 pixels for future work.

4.1.4 Fully out of sample evaluation

To evaluate the extent to which the performance is the result of overfitting on backgrounds, watermarks, repeated signers, and other effects of YouTube series, this paper also performs a fully out-of-sample independent evaluation. This evaluation selected 92 clips from the independent curated NCSLGR corpus for American Sign Language [24]. No signers or series overlap with SLANG-3k.

On this independent test set, the 128×128 and 231×231 temporal models scored 100%, and the 32×32 temporal models scored 97% in correctly classifying all clips as ASL, suggesting that the effects of repetition in the training and testing sets may not be a substantial driver of performance.

4.2. Qualitative evaluation

Figure 4 displays a set of sample validation images and the likelihood evaluations for each language. As indicated in these samples and the confusion matrices, the models struggle most to distinguish ASL and DGS. This could be a linguistic effect (ASL and DGS are more closely related to each other than either is to BSL), or it may be a dataset genre effect (the BSL data had a different distribution of genres), or it may be both.

Future work on identifying sign languages on the internet may benefit from introducing a neutral class of people who are not signing, given that some clips had no signer. For instance, the man at the far right in Figure 4 is not signing, yet the network classifies this image as BSL because the training data led it to expect an (unpictured) interpreter.

As part of qualitative evaluation, the author produced pairs of heatmaps through occlusion. By convolving a 24×24×5 white cuboid through the video volume with stride 8, this paper obtained estimates of the effect of each region on correct prediction (in green and red) and the importance of each region to differentiation between the top two classes (in blue), as in Figure 5 and Figure 6.

Formally, we calculate margins of certainty m and the changes over baseline c scores for each 24×24×5 region of the image, resulting in sixteen distinct measurements that are summed for each region of size 8×8×5:

$$c_i = p(y_i|\text{original image}) - p(y_i|\text{altered image})$$

$$m_i = p(y_i|\text{altered image}) - p(y_j|\text{altered image})$$

where p is the probability assigned by the model to that class and y_j is the class score of the runner-up. We then scale m to have range $[0, 1]$ and c to have range $[-1, 1]$ through the transformation:

$$f(x_i) = \frac{x_i}{\max_x |x|}$$

For the green and red change over baseline maps using c, we create a mask in which areas with positive scores are green and areas with negative scores are red, since positive values on c indicate that occlusion of the corresponding area decreases the correct score and negative values indicate that occlusion increases the correct score. The alpha channel is adjusted such that darker areas indicate a relatively stronger positive or negative effect.

For the blue margin of certainty maps using m, we create a mask in which areas with low scores are transparent and high scores are blue. In this setup, transparency indicates which regions are important for differentiation: when occlusion of a region produces a small difference between the correct class and the runner-up, then that region is important for being able to distinguish between the two classes.

We find that for many instances, performance is tied to actions suggestive of a language. For instance, in Figure 5, we find that the network relies on the highly BSL-indicative two handed fingerspelling and arm orientation. However, for videos in a series, the network sometimes finds the background more informative than the language, as in Figure 6.

An alternative way to understand whether the model is identifying linguistic similarities or other similarities is to

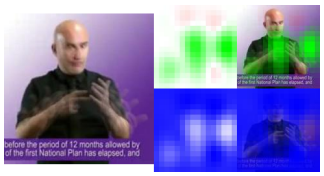

Figure 5. The signer and his BSL-indicative two-handed fingerspelling matter. Green regions are important to correct prediction; clear regions are important to differentiation between languages.

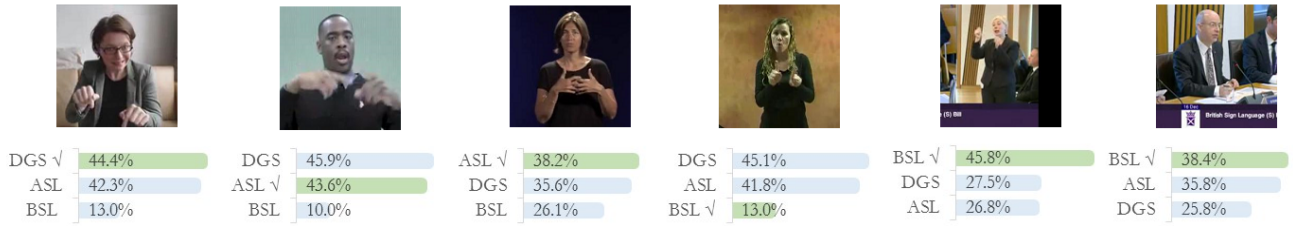

Figure 4. Example performance of 231×231×5 (big) model on validation set images. ASL and DGS are particularly confusable, as expected linguistically. Some images without signing (like the image at far right) are labeled as per an interpreter not in crop.

collect an additional set of clips from languages that vary in their similarity to ASL, BSL and DGS, and attempt to categorize them using the current model. One would expect that languages that are similar to ASL, like Ghanaian, Malaysian and Indonesian Sign Language would tend to be confused for ASL; that languages that are most similar to DGS, like French Sign Language, would be confused for DGS; that languages that are similar to BSL, like Australian and New Zealand Sign Language, would be confused for BSL; and that languages that are not related to ASL, BSL or DGS, like Turkish and Japanese Sign Language, might contextualize the importance of non-linguistic phenomena.

Given this analysis, we plot the confusion matrix using correspondence analysis in Figure 7. Correspondence analysis is a multivariate statistic technique that produces a visualization such that items with similar profiles appear near each other, and items with different profiles appear far from each other. To do so, it performs singular value decomposition $D = U\Delta V^T$ on a transformation of the contingency table that accounts for deviations from independence and for rarity of response. The full mathematical derivation is somewhat involved; the reader is referred to [12] for additional information.

We find in the correspondence analysis visualization that the ASL family (ASL, GHA, IDN, MAL) are coherently centered around ASL, and that that group connects very cleanly to the related languages of DGS and FRA. However, the separation between BSL, NZL and AUS is surprising, as these languages are very closely related [36]. A review of the videos used suggests that perhaps the NZL and AUS videos are in very simple NZL and AUS, such that someone familiar with ASL-DGS-similar languages could indeed understand the clips that are being categorized with that family. Alternatively or simultaneously, their positioning reflects the effect of noise in the data. Unfortunately, little additional interpretability is provided by the position of the Turkish and Japanese Sign Language clips.

Figure 6. The signer is unimportant compared to his infomercial background. Green regions are important to correct prediction; clear regions are important to differentiation between languages.

5. Conclusion

This work establishes that temporal information is important in classifying video clips as to the sign language being used. In particular, it finds that temporal information is more useful than better spatial feature extraction, and provides quantitative evidence that humans struggle to identify languages from still frames. This task therefore contrasts with previous action detection work in areas like sports, for which temporal information holds less importance. This paper releases SLANG-3k, a public benchmark dataset for sign language identification on the internet, and finds that there may be a limit around 66% accuracy given a sixth of a second of data.

The author recommends that future work focus on training deeper and more effective networks with longer sequences of frames as supported with better hardware and/or PCA to reduce the video data to a more reasonable size, and on the area of data augmentation (larger datasets, additional resizing, changes in lighting and color contrasts).

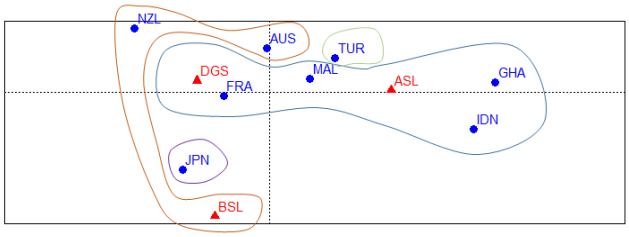

Figure 7. Correspondence analysis on additional languages; languages with similar profiles appear near each other and related languages are circled. No units appear because distances are not strictly meaningful or comparable in this PCA-like visualization with multiple data types. New languages use their country codes.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding* , 29-30. Springer, 2011.

[2] G. Bradski. The OpenCV Library. In *Dr. Dobbs Journal of Software Tools*, 2000.

[3] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 161-175, 1994.

[4] S. Chintala. matio: A LuaJIT FFI interface to MATIO and simple bindings for torch. (Torch library).

[5] A. Clark. Pillow 3.1.1 (Python library).

[6] R. Collobert, C. Farabet, K. Kavukcuoglu and S. Chintala. Torch: A scientific computing framework for LuaJIT.

[7] T. Dunning. Statistical identification of language. Technical Report MCCS 94-273, Computing Research Laboratory, New Mexico State University, 1994.

[8] A. Garg, V. Gupta and M. Jindal. Survey of language identification techniques and applications. In *Journal of Emerging Technologies in Web Intelligence*, 6(4), 2014.

[9] B. G. Gebre, P. Wittenburg, and T. Heskes. Automatic sign language identification. In *Proceeding of the 20th IEEE International Conference on Image Processing (ICIP)*, 2626-2630, 2013.

[10] B. G. Gebre, O. Crasborn, P. Wittenburg, P. Drude, and T. Heskes. Unsupervised feature learning for visual sign language identification. In *ACL 2014: The 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[11] R. Girshick, J. Donahue, T. Darrel, and J. Malik. Rich feature hierarchies for accuracy object detection and semantic segmentation. In *CVPR*, 2014.

[12] M. Greenacre. Correspondence analysis in practice. Champman & Hall, London, 2007.

[13] M. Greenacre, O. Nenadic and M. Friendly. ca: simple, multiple and joint correspondence analysis (R package).

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural Computation*, 9(8):1735-1780, 1991.

[15] 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 International Computer Vision and Pattern Recognition (CVPR 2014)* , 2014.

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. CoRR, 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. NIPS 2012: Neural Information Processing Systems, Nevada, 2012.

[18] H. Leuninger. Gebärdensprachen: Struktur, Erwerb, Verwendung. Buske, Hamburg, 2005.

[19] S. MacNamara, P. Cunningham, and J. Byrne. Neural networks for language identification: A comparative study. In *Information Processing and Management*, 34(4), 395-403, 1998.

[20] M. Maraqa, F. Al-Zboun, M. Dhyabat, and R. Abu Zitar. Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In *Journal of Intelligent Learning Systems and Applications*, 4(1), 2012.

[21] D. McKee and G. Kennedy. Lexical comparison of signs from American, Australian, British, and New Zealand Sign Languages. In *The signs of language revisited: an anthology to honor Ursula Bellugi and Edward Klima*, K. Emmorey and H. Lane (eds), 2000.

[22] P. Mekala, Y. Gao, J. Fan, and A. Davari. Real-time sign language recognition based on neural network architecture. In *IEEE 43rd Southeastern Symposium on System Theory*, 195-199, 2011.

[23] Q. Munib, M. Habeeb, B. Takruri, and A. Al-Malik. American Sign Language (ASL) recognition based on Hough transform and neural networks. In *Expert Systems with Applications*, 32, 24-37, 2007.

[24] C. Neidle and C. Vogler. A new web interface to facilitate access to corpora: Development of the ASLLRP Data Access Interface. In *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*. LREC 2012, Istanbul, Turkey, 2012.

[25] Numerous contributors. image: An image toolbox for Torch (Torch library).

[26] Numerous contributors. nn: Neural network package (Torch library).

[27] Numerous contributors. optim: Optimization package (Torch library).

[28] C. Oz and M. C. Leu. American Sign Language word recognition with a sensory glove using artificial neural networks. In *Engineering Applications of Artificial Intelligence*, 24(7), 1204-1213, 2011.

[29] M. Padro and L. Padro. Comparing methods of language identification. In *Proceedings fo the XX Congreso de la Sociedad Espanola para el procesamiento del Lenguage Natural*, 2004.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, In *CoRR*, 2014.

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *CoRR*, 2013.

[32] E. Singer, P. A. Torres-Carrasquillo, T. P Gleason, W. M. Campbell and D. A. Reynolds. Acoustic, phonetic and discriminative approaches to automatic language identification. In *Proceedings Eurospeech*, 9, 2003.

[33] R. Sutton-Spence and B. Woll. The Linguistics of British Sign Language: An Introduction. Cambridge University Press, 1998.

[34] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object location using convolutional networks. CoRR, 2014.

[35] P. Toman and A. Kuefler. Classifying non-manual markers in American Sign Language. Stanford University, CS 229 Class Project, 2015.

[36] L. Trevilla. *Ethnologue: Languages of the World*, SIL International, 2015.

[37] C. Valli, C. Lucas, and K. J. Mulrooney. *Linguistics of American Sign Language*, Gallaudet University Press, 2005.

[38] E. Wilson and G. Ansprach. Neural networks for sign language translation. In *SPIE: Applications of Artificial Neural Networks*, 4, 1993.

[39] M. A. Zissman. Comparison of four approaches to automatic language identification of telephone speach. In *IEEE Transactions on Speech and Audio Processing*, 4(1), 31-44.