Large-scale Learning of Sign Language by Watching TV (Using Co-occurrences)

Tomas Pfister[1]
tp@robots.ox.ac.uk
James Charles[2]
j.charles@leeds.ac.uk
Andrew Zisserman[1]
az@robots.ox.ac.uk

[1] Department of Engineering Science
University of Oxford
Oxford, UK

[2] School of Computing
University of Leeds
Leeds, UK

Abstract

The goal of this work is to automatically learn a large number of signs from sign language-interpreted TV broadcasts. We achieve this by exploiting supervisory information available in the subtitles of the broadcasts. However, this information is both weak and noisy and this leads to a challenging correspondence problem when trying to identify the temporal window of the sign.

We make the following contributions: (i) we show that, somewhat counter-intuitively, mouth patterns are highly informative for isolating words in a language for the Deaf, and their co-occurrence with signing can be used to significantly reduce the correspondence search space; and (ii) we develop a multiple instance learning method using an efficient discriminative search, which determines a candidate list for the sign with both high recall and precision.

We demonstrate the method on videos from BBC TV broadcasts, and achieve higher accuracy and recall than previous methods, despite using much simpler features.

1 Introduction

TV programmes in many countries across the world are now routinely broadcast with both subtitles and an overlaid signer translating to the Deaf audience. Our aim is to exploit this material to learn *signs* corresponding to English *words* in the subtitles [3, 5]. Within the UK alone, over five hours of sign language-interpreted TV programmes are broadcast every day by the BBC. This data provides a continuous and rich source of training material for learning sign language.

Our long term vision is to build a database of word-sign pairs for a large number of signs and signers that can be used to train a large-scale person-independent sign language to text translator. This translator could then be used by some of the 70 million Deaf worldwide to communicate with people who don't understand sign language.

Our objective in this paper is to build the database, or in more detail: given an English word, to automatically and accurately obtain the video of the sign corresponding to that word. The mechanism for learning the sign is the following: the English word is used to select a set of subtitles and associated videos that contain the word – the positive sequences – and a set of subtitles and videos that do not – the negative sequences [3, 5]. The video of the sign is learnt from this training material. This is a very challenging correspondence problem

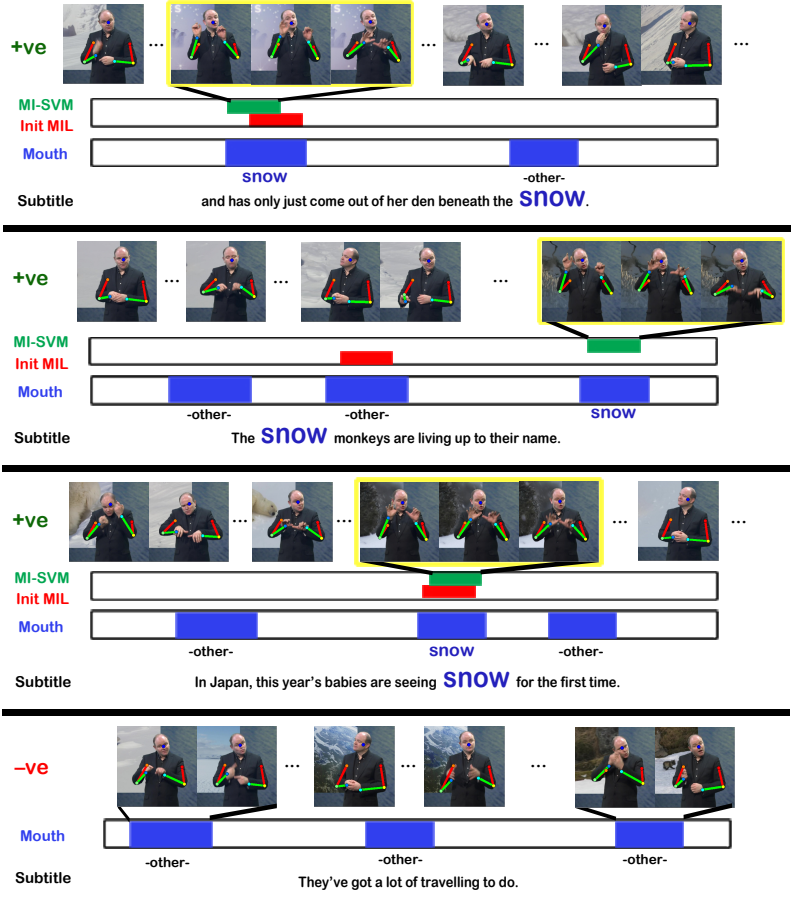

Figure 1: **Learning signs from co-occurrences of subtitle text, mouth and hand motion.** The top three rows are positive subtitle sequences which contain the text word and sign for 'snow'. The final row is an example of a negative subtitle sequence which does not contain 'snow'. Signs are learnt from this weakly aligned and noisy data. A fixed size temporal window is slid across the frames in which mouth motion occurs (blue). The rest of the sequence can be ignored, thus reducing the temporal search space. Candidate signs are proposed by a discriminative MIL search using temporal correlation. A subset of these candidates (red) are used to initialise a MI-SVM, resulting in the final correspondence matches (green). The red and green lines on the signer show the detected limbs and head.

as the subtitles are not temporally aligned with the signs – a sign (typically 8–13 frames long) could be anywhere in the overlapping subtitle video sequences (typically 400 frames). Furthermore, there is not a one-to-one mapping between signs and English words, and the occurrence of a word in the subtitle does not always imply that the word is signed. Thus the supervision provided is both weak and noisy.

In this paper we demonstrate that *co-occurrences* can be used to significantly improve the 'signal-to-noise' ratio for this problem. In particular, we demonstrate that in sign language, co-occurrences of *lip and hand motion* can be used to substantially narrow down the correspondence search space – signers often mouth the word that they are signing, an important cue that has been overlooked in previous work to the best of our knowledge. This idea is not limited to sign language: co-occurrences of lip motion and speech can be used to aid speech recognition [2, 9, 21], and the vicinity of objects can be used to aid human action recognition [23] (*e.g.* a phone close to human and a pose with a hand close to the head can be used to detect that the action is 'phoning').

We cast the problem as one of Multiple Instance Learning (MIL) [7, 18], where the training data are visual descriptors (hand trajectories) with weak supervision from subtitles. We proceed in three steps: (i) the search space for correspondences is significantly reduced by exploiting lip and hand motion co-occurrences to filter away irrelevant intervals of the temporal sequences (Section 2); (ii) candidates for the signs are obtained using an efficient discriminative search over all remaining sequences (Section 3); and finally (iii) these candidates are then selected or rejected using the MIL support vector machine framework (MI-SVM) [1]. Figure 2 shows the processing pipeline of the sign learning algorithm.

The building blocks for the method are a state-of-the-art automatic real-time upper-body tracker and an accurate mouthing classifier (Section 4). We demonstrate that with the reduction in search space and discriminative learning, quite simple features are sufficient to successfully extract the signs – in particular we do not need to represent hand shape at this stage. As will be seen in Section 5, we achieve superior results to previous work at a much lower computational cost.

In previous work, Farhadi and Forsyth [12] considered the problem of aligning an American Sign Language sign with an English text subtitle, but under much stronger supervisory conditions than more recent approaches. The closest work to ours is Buehler *et al*. [3] who used similar weak and noisy supervision from subtitles. However, their method does not exploit mouth motion, and relies on performing a computationally expensive brute force search over all temporal windows – here we avoid both the exhaustive search and also the necessity to represent hand shape and orientation as they did. Cooper and Bowden [5] used a temporally constrained adaptation of apriori data mining on hand and head positions to learn signs. Their method is a complementary method to ours and provides another way to select candidates. Other approaches have typically required manual training data to be generated for each sign [6, 8, 15, 20, 25, 26, 27], *i.e*. a signer performs each sign in laboratory conditions with manual labelling of the signs afterwards – a labour-intensive and expensive process.

2 Shrinking Search Space using Mouthing Co-occurrences

A key contribution of this paper is the discovery that mouth patterns are very helpful for aligning signs. This is because the Deaf in most countries commonly use their mouth to express the lip pattern of the English (or other written/spoken language) equivalent word that they are signing, also known as 'mouthing'. This mouth information is valuable in two distinct ways: (i) knowing that the Deaf mouth the word in the majority of signs, one can discard frames where the mouth is not open, and thus considerably narrow down the search space when matching signs; and, to a lesser extent, (ii) the similarity of lip patterns across repetitions of the same sign can be used as an additional cue for matching different instances of the same sign.

In order to use the mouthing information to discard frames where the signer is not speaking, we train a per-frame classifier for predicting 'speaking' vs 'not speaking' (details in Section 4.3). The output of this classifier is used to prune the search space (effectively filtering away frames in which the signer is predicted not to be mouthing). The remaining search space is marked blue in Figure 1.

This results in a substantial decrease in the number of temporal windows that need to be considered for correspondences. For example, in the sequences of Figure 1, the search space is $l = 400$ frames, and sign correspondences are searched for with a fixed-size temporal window of $w = 13$ frames. Without using mouthing information, this would yield $n = l - w + 1 = 388$ candidate temporal windows per positive sequence. However, by cutting this search space down to three $l = 25$ frame windows using mouthing information, the number

Preprocess all videos
for each frame
– segment signer and locate signer's head and hands *(Sect 4.2)*
– classify mouth as open/closed and extract mouth SIFT descriptor *(Sect 4.3)*
➡ **output:** hand and head positions, mouth open/closed probability and mouth SIFT descriptor

For a particular word, e.g. 'snow'
– find positive & negative sequences using subtitles *(Sect 4.1)*
– obtain temporal windows by sliding a fixed-size window over frames with an open mouth in each sequence *(Sect 2)*
– extract feature vector for each temporal window *(Sect 4.4)*
– find sign candidates using discriminative MIL search *(Sect 3.1)*
– train a MI-SVM classifier with initial candidates *(Sect 3.2)*
➡ **output:** instances of the sign 'snow'

Figure 2: Sign extraction pipeline.

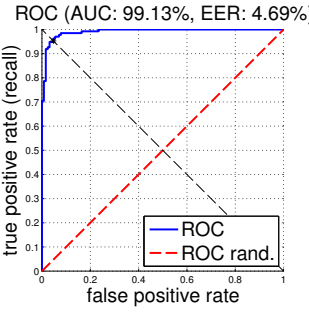

Figure 3: ROC curve of the person-independent mouthing classifier.

of candidate windows drops 90% to 39. This order-of-magnitude reduction in search space not only improves the 'signal-to-noise' ratio in the correspondence search but considerably speeds up the search.

3 Automatic Sign Extraction using MIL

Given a target word occurring in the subtitles, the aim here is to extract examples of the corresponding sign. The key idea is to search for signs that are common across the positive sequences (the sequences where the target word occurs in the subtitles), but uncommon in negative sequences (where the target word doesn't occur in the subtitles). Since the positive labels are on a subtitle sequence level rather than on a window level, the task can naturally be formulated as a Multiple Instance Learning (MIL) [7, 18] problem as shown in Figure 1. MIL is a variation of supervised learning for problems that have incomplete knowledge about the training set's labels. Unlike supervised learning in which each training instance has a label, in MIL the labels are on a *bag* level, where each bag consists of *instances*. If a bag is *positive*, then *at least one instance* in the bag is positive. If it is *negative*, then *no instance* in the bag is positive. In our scenario, the bags are the sequences, and the instances are features computed from fixed sized temporal windows within the temporal intervals in which the signer is mouthing. Positive bags are from positive sequences, and negative bags from negative sequences. Details of features are given in Section 4.4.

As an example, the word 'snow' occurs in 30 subtitles, thus yielding 30 positive sequences, each around 400 frames long. Out of these 400 frames, on average six subsets (each 30 frames long, examples shown in blue in Figure 1) contain mouthing. With a temporal window size of 13, this yields 108 temporal windows for each positive sequence, or in total 3,240 temporal windows for all positive sequences. However, an additional challenge is posed by the fact that 'snow' is only signed in 10 out of the 30 sequences. In this case, our task is to find the 10 out of 3,240 temporal windows that contain the target sign. With a 'signal-to-noise' ratio of less than 0.4%, this is a very challenging problem even when using mouthing to cut down the search space (in this case, mouthing reduces the number of temporal windows from 11,640 to 3,240).

The MIL proceeds in two stages: (i) finding good candidates for the target sign temporal windows using temporal correlation scores, and (ii) refining this 'candidate list' using MI-SVM. Example outputs from the two steps are visualised in Figure 1.

3.1 Multiple instance learning using discriminative search

The method for finding candidate temporal windows relies on computing *temporal correlation scores* between the temporal windows. The input is a set of feature vectors $\{x_i\}$, each representing the temporal motion of the hands and mouth over a fixed sized temporal

window. Each vector x_i is composed of blocks covering aspects of the signing of a given temporal window, such as lip motion and distance between the hands. The vector is normalised such that the dot product $x_i \cdot x_j$ between two such vectors, x_i and x_j, gives the *temporal correlation score* of the 'signals' (lip motion, hand motion) over the two temporal windows (Section 4.4 gives the details). The temporal correlation measures how similar the hand and lip trajectory is between the two windows, with a value of 1 indicating perfect correlation, and -1 indicating anti-correlation.

The task is to determine for each x_i how likely it is to be the target sign. This is accomplished by using each x_i to classify the positive and negative sequences. The idea is that if x_i is actually the sign then its correlation (i.e. $x_i \cdot x_j$) with some vector x_j in a positive sequence will be higher than with any vector x_k in a negative sequence. To this end, for each x_i all sequences are ranked using the 'classifier' score $x_i \cdot x_j$, and the performance of the classifier is assessed using the area under its ROC curve (AUC). For the purpose of annotating the vectors when computing the ROC curve, any vector in a positive sequence is deemed positive, and any vector in a negative sequence deemed negative. A good candidate x_i will rank the positive sequences first, and thus have a higher AUC, than a poor candidate. Note, the annotation of 'positives' here is noisy, since only a fraction of the windows in positive sequences contain the sign.

In summary, we measure the 'quality' of each candidate temporal window using the AUC of its ROC curve – a form of one shot-learning. The windows are then ranked according to their AUC scores, and this ranked list is used below for initialising and training the MI-SVM.

Discussion. Before temporal correlation scores are computed, the features x is whitened to $\hat{x} = \Sigma^{-1/2}(x-\mu)$, where Σ is the cross-correlation matrix, x is an L_2-normalised input feature vector and μ are the means of the input feature vector. Whitening effectively 'equalises' the features, thus making feature variations more comparable, leading to better learning. The above initialisation method is equivalent to training an exemplar Linear Discriminant Analysis (LDA) classifier per temporal window. This is because whitening of the feature space before computing temporal correlation scores is equivalent to exemplar LDA [13, 14].

This MIL initialisation method is not limited to applications in sign language – the same idea can be used to initialise MIL in other weakly supervised tasks.

3.2 Temporal correlation based MI-SVM

In MI-SVM, given the positive and negative bags as input, a classifier w is learnt to select the positive instances x from the positive bags by an algorithm that alternates between: (i) selecting the positive instance in each bag as those with maximum score $w \cdot x$, and (ii) standard SVM training using the selected positives and all negative instances. More formally, given a set of input temporal windows $x_1, \ldots, x_n$ grouped into bags $\mathbf{B}_1, \ldots, \mathbf{B}_m$ according to which positive/negative video subsequence they belong to, where each bag $\mathbf{B}_I$ is associated with a label $Y_I \in \{-1, 1\}$, we optimise

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \xi_I \tag{1}$$

$$\text{s.t.} \quad \forall I : Y_I \max_{i \in I}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_I, \quad \xi_I \geq 0 \tag{2}$$

where i are indices for instances and I are indices for bags.

However, a good initialisation is necessary for the algorithm to succeed. The ranked candidate list from Section 3.1 is used to first initialise and then train the MI-SVM. In detail, the shortlist uses the top 20% highest-ranked candidate windows as positives. The shortlist candidates are allocated to their positive sequences and a weight vector w is learnt using the

MI-SVM algorithm, where the top ranked candidates in each sequence are used for initialisation. The candidates with maximum SVM score in each sequence form the final result. A separate MI-SVM is trained and evaluated for each target word.

Training in this manner means that the weight vector is learnt from instances with far fewer false positives (higher 'signal-to-noise'). It is demonstrated in Section 5.2 that this substantially improves its performance (compared to MI-SVM learning from all windows directly). Figure 1 shows the selection 'in action' on three positive subtitle sequences.

Discussion. The weight vector w is learnt discriminatively and thus can learn to suppress part of the feature vector. For example, if the distance between the hands carried no discriminative information for a particular set of positive sequences, then this block of the feature vector need not be selected. The vector w is a stronger classifier than the exemplar LDA classifier above, since w uses multiple positive samples for training, rather than being constructed from only a single sample.

4 Implementation Details

This section describes the implementation details of how training data (positive/negative sequences, and a feature vector based on upper-body joint locations and mouth features) are automatically generated from subtitles and video material.

4.1 Text processing

Each subtitle text consists of a short text, and a start and end frame indicating when the subtitle is displayed. The subtitles are stemmed (common inflections *e.g.* "s", "ed", "ing" are removed) and stop words are filtered away.

Positive sequences. A positive sequence is extracted for each occurrence of the target word in the subtitles. Since the alignment between subtitles and signs is very imprecise due to latency of the signer (who is translating from the soundtrack) and differences in language grammar, some 'slack' is padded to the sequence window. Given a subtitle where the target word appears, the frame range of the positive sequence is defined as from the start of the *previous* subtitle to the end of the *next* subtitle. This results in sequences of about 400 frames in length. In contrast, signs are generally 7–13 frames long.

Negative sequences. Negative sequences are extracted by searching for subtitles where the target word *does not* appear. This yields on average about 100,000 negative temporal windows per video.

4.2 Large-scale human co-segmentation and joint tracking

In previous work [22] we developed a fully automatic arm and hand tracker that detects joint positions over continuous sign language video sequences. However, the method relies on accurate signer foreground segmentations, which are challenging to determine since the colours of the foreground and background are often similar. In this work we improve upon this segmentation method by using additional freely available information, namely that the same TV programmes are broadcast with and without an overlaid sign language interpreter, as shown in Figure 4. If the two videos can be perfectly aligned, and any noise can be removed, then this provides a very strong cue for segmenting the foreground.

This is however not straightforward as the two videos differ greatly in broadcast quality (high definition vs standard definition), which results in many spurious edges in the difference image. We tackle this by finding edges in the original video and filtering these away from the difference image. The difference image then undergoes a set of image processing operations that produce a clean foreground 'clamp region' shown in Figure 4(c). This

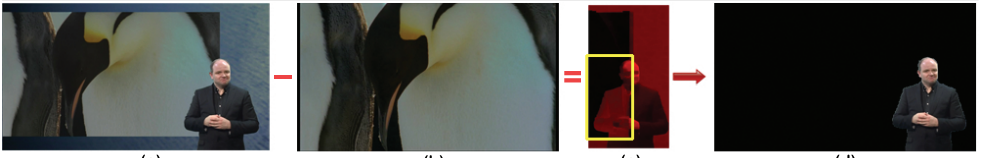

Figure 4: **Large-scale human co-segmentation.** The signer-overlaid frame (a) and original frame (b) are subtracted after alignment, resulting in a difference image (c) that is used as a foreground clamp (in yellow) and to generate GrabCut constraints. (d) shows the final result.

per-frame clamp region is used (i) as a foreground clamping region in GrabCut [24], (ii) for building an accurate video-wide global foreground colour model, and (iii) for partially replacing the colour posterior unary in frames with similar foreground and background colours. A secondary GrabCut unary is also computed based on the background colour model of the video without an overlaid signer.

These improvements yield near-perfect segmentations similar to Figure 4(d) for all signed TV broadcasts, which significantly reduces the search space when tracking the signer's upper-body joints. Example co-segmentation results are available online.

As in Pfister *et al*. [22], the segmentation is used to 'cut out' the signer from the video. A colour posterior that gives the likelihood of each pixel in the segmented region being skin/torso/background is then computed. This feature representation is used as the input to a Random Forest classifier, which is trained on hours of tracking output from an accurate, but slow and semi-automatic, tracker [4]. This yields an automatic real-time upper-body tracker that estimates joints accurately on unseen signers. The output we use is the position of the head and hands in every frame.

4.3 Mouthing classifier

Facial landmarks are detected using the method of Everingham *et al*. [10]. A similarity transform is then applied to the mouth feature points to yield a scale, rotation and translation normalised mouth patch. A binary Chi-squared kernel SVM is trained to classify each such patch as mouthing / non-mouthing using Local Binary Pattern (LBP) [19] features with cell size 8 extracted from the mouth patch of size 32×52 pixels. The dimensionality of the feature vector is 1,392 per frame. At test time the SVM output scores are thresholded to yield windows in which mouth motion is detected (blue areas in Figure 1). Training details and performance are given in Section 5.2.

4.4 Feature vector computation

For each frame we have the position of the signer's head and hands (from Section 4.2), and a descriptor for the mouth shape (a 128-dimensional SIFT descriptor [17]). For each temporal window, a number of feature blocks are computed based on the trajectory of the hands: (i) the relative (x,y) coordinates of each hand compared to the position of the head, (ii) the differences $(x_{\text{right}} - x_{\text{left}}, y_{\text{right}} - y_{\text{left}})$ in coordinates between the two hands, and (iii) a vector describing the direction and magnitude of motion between the first and last frame in the temporal window: $(x_{\text{last}} - x_{\text{first}}, y_{\text{last}} - y_{\text{first}})$. In addition, the feature vector contains a block for the SIFT descriptor of the mouth patch for each frame in the temporal window. Each feature block is then L_2 normalised so that $x_i \cdot x_j$ becomes the temporal correlation between two temporal windows with feature vectors x_i, x_j.

The feature dimension per frame is 134 out of which 128 is a SIFT describing the mouth, and the remainder describes the joints. For a temporal window size of 13, the total feature vector dimensionality is 1,746 (including two vectors giving the direction of motion between the first and last frame of each hand). The entire feature vector is a concatenation of all the L_2 normalised blocks, and this is then L_2 normalised.

5 Experiments

First the dataset and evaluation measure are described (Section 5.1); then the performance of the mouthing classifier and the helpfulness of mouthing information is evaluated (Section 5.2); and finally the results are compared to the state of the art (Section 5.3). Sample videos are available online.[1]

5.1 Dataset, evaluation measure and computation time

The sign extraction dataset consists of 35 high-definition TV broadcast videos, with 17 different signers, and in total 30 hours of data. Each video typically contains between 40K and 85K frames of sign-interpreted video content from a variety of BBC TV programmes. All frames of the videos are automatically assigned segmentations, joint labels and mouthing scores using the methods described in Section 4.2 and 4.3.

The 1,000 most frequently occurring words in the subtitles are selected, and the algorithm of Section 3 is used for each of these to extract the corresponding signs.

Manual ground truth. A set of 41 subtitle words (animal, antique, asian, bank, beacon, bear, beautiful, beef, bike, blood, buy, chinese, chocolate, epigenome, fake, feel, gram, heart, heat, industry, jelly, jewish, kill, market, milk, pay, reindeer, rugby, school, science, sell, simple, snow, song, sound, target, vision, war, winter, work, year) are selected at random from the 1,000 most frequently occurring words, and for these the ground truth sign temporal windows are annotated for all positive sequences for that word. The number 41 is chosen for comparison purposes, as Buehler *et al*. [3] annotated the same number of words.

Mouthing classifier train/test sets. The mouthing SVM classifier is trained on mouth LBPs of five unseen signers that are not in the sign extraction dataset. Mouths were manually annotated as either open or closed in 800 frames for each signer (4,000 frames in total). Testing for the mouthing classifier is conducted on three randomly chosen signers in the sign extraction set, each with 200 manually annotated frames (600 frames in total).

Evaluation measures. Given an English word, the goal is to identify many examples of the corresponding sign. We use two evaluation measures: *sign-level* (coarser) and *instance-level* measures. In the sign-level measure used by Buehler *et al*. [3], the output is deemed a success if at least 50% of retrieved candidates (maximum one per subtitle sequence) show the true sign (defined as a temporal overlap of at least 50% with ground truth). In the instance-level measure, we report precision and recall computed per word and then averaged across words. Precision measures the percentage of retrieved windows that contain the correct sign, whereas recall measures the percentage of sign instances that are retrieved.

Computation time. The following computation times are on a single core of a 2.4GHz Intel Quad Core I7 CPU. Segmentations, joints and mouthing classifier scores for one frame are computed in 0.3s (3fps). The runtime for the MIL initialisation step is on average 20s per subtitle word, and MI-SVM converges on average in 1min 30s, totalling 1min 50s per word. Initially there are on average 4,000 temporal windows, of which the MIL initialisation step returns around 800 as a shortlist.

5.2 Mouthing classifier and advantages of using co-occurrences

In this subsection the performance of the mouthing classifier is evaluated, and the advantages of using mouthing for sign extraction are explored.

Mouthing classifier. Figure 3 shows the ROC curve for the mouthing classifier when trained and tested on different signers as described in Section 5.1. As the ROC curve demonstrates, the classifier gives a quite reliable measure of whether the mouth is open or closed. On average, 71.2% of the search space is discarded using this method.

[1] http://www.robots.ox.ac.uk/~vgg/research/sign_language

Advantages of using co-occurrences. The instance-level average precision is 57.1% and recall is 78.0%. With the sign-level evaluation measure, the performance is 92.7%. We achieve good results for a wide variety of signs: (i) signs where the hand motion is important (*e.g.* 'snow'), (ii) signs where the hand shape is particularly important (*e.g.* 'jewish' where the hand indicates an imaginary beard), and even (iii) signs which are performed in front of the face (*e.g.* 'pork'), which makes detecting mouth motion difficult. MI-SVM suppresses the mouth part of the feature vector by assigning it a lower w weight.

The importance of the components and stages of the algorithm is evaluated next. If the mouthing classifier is not used for cutting down the search space, some signs can be detected, but the overall results are much poorer. This is reflected in the instance-level evaluation measure: average precision over 41 words drops to 17.8% and recall to 39.3%. If the initialisation step is omitted, and MI-SVM is instead initialised with all windows from the positive sequences (after search space reduction), results drop to 5.7% precision and 2.5% recall. We have also qualitatively evaluated our method over the 1,000 words. For more than half of the words, our method returns the correct sign one or more times in the top 10 final temporal windows selected by MI-SVM.

5.3 Comparison to previous publications

Direct comparisons to previous works are not possible since we do not have access to the same TV programmes with the same signs performed by the same signer. Moreover, previous work used standard-definition TV broadcasts, where the resolution was not good enough to detect facial feature points reliably. However, we show that our results are competitive when performing similar experiments to those in Buehler *et al.* [3] with a similar-sized vocabulary.

Using the sign-level evaluation measure, our 92.7% success rate far exceeds Buehler *et al.*'s rate of 78% on the same number of signs. However, we must also point out that this measure is only concerned with recall, and not precision. In fact, although our method has a good precision, it has lower precision (*i.e.* higher false positive rate) on the 41 word test set. This is only to be expected given that we are using far less discriminative features than [3] who also use hand shape and hand orientation. However, our method is much simpler both in terms of features and learning framework, and is extremely fast ($\approx$2min/word).

Cooper and Bowden [5] detect the signs for 53.7% of 23 words in a 30 min TV broadcast. The results are not directly comparable as different performance measures are used, but we detect 92.7% of nearly twice as many words at a fraction of the computational cost.

6 Conclusion

We proposed a framework for automatically learning a large number of signs from sign language-interpreted TV broadcasts. Our method exploits co-occurrences of mouth and hand motion to substantially improve the 'signal-to-noise' ratio in the correspondence search. Moreover, we proposed a principled method for initialising the correspondence search that significantly improves performance. We achieve superior results to those in previous work [3] with much simpler features and a much lighter learning framework. The ideas of exploiting co-occurrences and obtaining MIL temporal correlation candidates by discriminative learning from a single exemplar window could be applied to a variety of fields where weak supervision is available, such as learning actions [16], gestures and names of TV characters [11].

Acknowledgements: We are grateful to Patrick Buehler, Relja Arandjelovic, Karen Simonyan and Yusuf Aytar for discussions. Financial support was provided by Ehrnrooth Foundation, Kaute and the EPSRC grant EP/I012001/1.

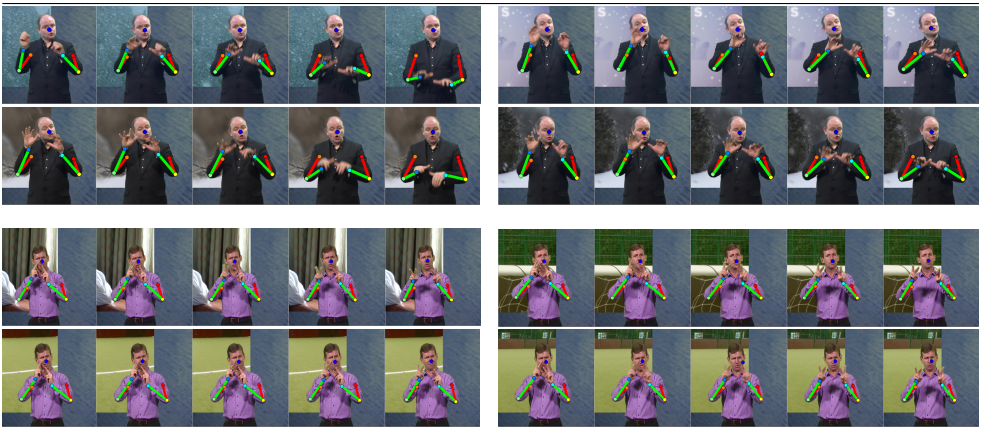

Figure 5: **Example sequences for the signs "snow" (top) and "vision" (bottom)** performed by two different signers and learnt automatically.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[2] C. Bregler and Y. Konig. "Eigenlips" for robust speech recognition. In *ICASSP*, 1994.

[3] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

[4] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Upper body detection and tracking in extended signing sequences. *IJCV*, 2011.

[5] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. CVPR*, 2009.

[6] H. Cooper, N. Pugeault, and R. Bowden. Reading the signs: A video based sign dictionary. In *Proc. ICCV Workshops*, 2011.

[7] T. Dietterich, R. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997.

[8] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, and H. Ney. Tracking using dynamic programming for appearance-based sign language recognition. In *FG*, 2006.

[9] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE MM*, 2000.

[10] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC*, 2006.

[11] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 2009.

[12] A. Farhadi and D. Forsyth. Aligning ASL for statistical translation using a discriminative word model. In *Proc. CVPR*, 2006.

[13] M. Gharbi, T. Malisiewicz, S. Paris, and F. Durand. A gaussian approximation of feature space for fast image similarity. *MIT Tech Report*, 2012.

[14] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. ECCV*, 2012.

[15] T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proc. BMVC*, 2004.

[16] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[17] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[18] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998.

[19] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 2002.

[20] S.C.W. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE PAMI*, 2005.

[21] E. Petajan, B. Bischoff, D. Bodoff, and M. Brooke. An improved automatic lipreading system to enhance speech recognition. In *SIGCHI*, 1988.

[22] T. Pfister, J. Charles, M. Everingham, and A. Zisserman. Automatic and efficient long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC*, 2012.

[23] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE PAMI*, 2012.

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *Proc. ACM SIGGRAPH*, 2004.

[25] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk- and wearable computer-based video. *IEEE PAMI*, 1998.

[26] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proc. ICCV*, 1998.

[27] C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel American sign language recognition. In *International Gesture Workshop*, 2004.