SIGN LANGUAGE TRANSCRIPTION WITH MACHINE LEARNING

Andrea Tan Kai Xuan[1], Ip Hei Man[2], Woo Chin Jian[2], Shen Bingquan[2]

[1]Nanyang Girls' High School, 2 Linden Dr, Singapore 288683

[2]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract

This paper investigates the feasibility of using classical Machine Learning (ML) methods to classify high-dimensional sign language videos into their corresponding gloss words (referred to here as "target words"). By analysing the application of ML on a gloss-free sign video dataset, a classification report is produced, which hints at the effectiveness of ML in classifying the target words based on their embeddings. Having obtained an **F1 score** of **97.49%** after training a gloss-free model with Random Forest, this paper aims to prove that classical ML methods are sufficient in carrying out video-to-gloss Sign Language Transcription (SLT) accurately, without the need for deep learning methods. This emphasises the potential of ML in SLT, fostering inclusivity for the Deaf and Hard-of-Hearing community.

1 Introduction

Sign languages are widely used by over 72 million people within the Deaf and Hard-of-Hearing community globally [1]. By making use of facial expressions, body movements, and hand gestures, sign language conveys meaning and emotion from one person to another. Though not universal, sign language can transcend spoken language barriers, promoting inclusivity. Leveraging Artificial Intelligence helps bridge communication gaps between these communities, and the wider society. This includes SLT, which converts sign language into text, facilitating seamless communication.

This paper first examines the strengths and limitations of both the gloss-based and gloss-free SLT approaches, before attempting to reproduce results from recent works by Ye et al. [2]. Thereafter, classical ML methods are applied to a gloss-free dataset, to assess if ML alone can effectively perform video-to-gloss transcription.

2 Existing Sign Language Transcription Methods

2.1 Gloss-based Methods

Gloss-based SLT methods include **Sign Recognition Pretrained (SRP)** [3] and **Self-Mutual Knowledge Distillation (SMKD)** [4]. SRP uses the PHOENIX-2014T dataset [3], which contains sign videos by 9 different signers, featuring a vocabulary of 1066 unique signs, as well as German translations of 2887 different words which were automatically transcribed then manually verified and normalised. This dataset was compiled by professional sign language interpreters and glosses were annotated by deaf specialists [3]. Building upon the SRP method, SMKD incorporates a shared classifier that aligns the outputs of the visual and contextual modules at gloss level, ensuring that they complement each other rather than to function independently. Gloss segmentation is also introduced to break down sign videos into segments corresponding to each gloss, thereby enhancing feature representation [4].

Glossing refers to the practice of representing each morpheme in sign language using written words or phrases that best capture its meaning, mapping a sign gesture to its corresponding gloss. Through a pipeline approach, sign language videos are converted to pose, then pose-to-gloss, and finally gloss-to-text. This one-to-one mapping method offers a simplified method of translating sign language, as it omits the complex nature of sign language, such as facial expressions.

However, creating large datasets with gloss annotations is both time- and energy-consuming, as it involves manual annotations of every gesture present in a sign language video. To ensure the accuracy of the gloss annotations is maintained, the manpower and expertise of professional sign language interpreters and translators is also required. This further adds to the cost of SLT, which limits it from being able to be widely implemented. Additionally, gloss-based methods may also result in over-simplification, or loss of the grammatical structure of sign language.

2.2 Gloss-free Methods

Gloss-free methods of SLT include the **Two-Stream Inflated 3D ConvNet (I3D)** [5] and **Visual-Language Pretraining (VLP)** [6]. In the I3D model, filters and pooling kernels of the 2D ConvNet are inflated into 3D. This allows the model to process seamless spatio-temporal feature extractors from sign language videos [5]. As for VLP, a 2-stage gloss-free approach is carried out to bridge visual-textual gaps using Contrastive Language-Image Pretraining (CLIP) and masked self-supervised learning, followed by the application of an encoder-decoder architecture with pre-trained models [6].

Unlike gloss-based methods, gloss-free methods do not rely on the use of gloss annotations, hence preserving the linguistic integrity of sign languages. ML models are trained to identify patterns and relationships directly from raw data (i.e. sign language videos). This approach captures the full complexity of sign language, allowing for greater flexibility of the model, which can perform well even on new unseen data.

However, without the intermediate step of gloss annotations, gloss-free methods tend to be of lower accuracy. This lies in the difficulty distinguishing between subtle variations in sign gestures that could be closely represented in a feature space. To address this problem in gloss-free methods, previous works have introduced strategies like Contrastive Learning [7], an image augmentation tool using self-supervised learning, and Gloss Attention [8], an attention mechanism that focuses on video segments in the same local semantic unit.

3 Representation Density Problem

In recent years, there have been great advancements in neural networks for SLT, such as the works by Moryossef et al. [9], Zhou et al. [10], and Lin et al. [11]. Yet, current SLT models still lag behind spoken language translations in terms of quality. This problem can be attributed to several factors, including the limited data available for ML models to train on, as well as the need to bridge the gap between natural language processing and computer vision.

As a visual language, sign language faces the **Representation Density Problem.** That is, hand gestures that appear visually similar are also closely represented in the feature space. This would lead to the misinterpretation of these gestures by automated sign language translators, confusing words with similar visual representations, despite them having distinct meanings.

Especially prominent in gloss-free SLT, models face difficulty in learning semantic boundaries in continuous sign videos, contributing to translation ambiguity [2].

To prove the existence of the Representation Density Problem, **T-distributed Stochastic Neighbor Embedding (t-SNE)** [12] is implemented in this paper. t-SNE is a statistical tool that enables high-dimensional data to be visualised in a lower-dimensional space. In this context, t-SNE is used to reduce the dimensionality of sentence embeddings to a two-dimensional space, all while preserving the local relationships between the data points. By implementing t-SNE, the gloss-free dataset can be plotted on a scatter plot, with target words colour-coded. This visualisation helps to identify patterns such as distinct clusters, if any, so as to better understand and confirm the existence of the Representation Density Problem.

4 Methodology

4.1 Choice of Dataset Used

The **I3D model** was used, which was pre-trained on DeepMind Kinetics [13], which consists of 400 classes of human actions with 400 unique YouTube clips per class. After learning the 3D features from the pre-training stage, I3D was fine-tuned using ChaLearn249 IsoGD [14], a dataset specific to Sign Language Recognition [15]. Since I3D consists of 3D kernels inflated from 2D convolutional kernels, it can process spatial (frame-level) and temporal (across frames) information simultaneously. This makes I3D well-suited for SLT, given its efficiency in tasks concerning motion dynamics, such as the gestures in sign language.

A gloss-free dataset was chosen over a gloss-based dataset as it would be more challenging to classify video embeddings by their target words, without the intermediate step of gloss annotations. If ML methods can perform well on even a gloss-free dataset, it further demonstrates that ML is sufficient for carrying out video-to-gloss SLT without relying on gloss annotations.

4.2 Evaluating Sign Density Ratio

Apart from using t-SNE to validate the Representation Density Problem, we computed a quantitative metric to measure the density of each gloss, G_i. The **Sign Density Ratio (SDR)** is defined as the ratio of the Intra-Gloss Distance (distance within a single gloss) to the average Inter-Gloss Distance (distance of a gloss to all other glosses).

A lower SDR value indicates more distinct feature representations, reducing the likelihood of encountering the Representation Density Problem discussed in Section 3 and this is usually observed in gloss-based methods. On the other hand, a higher SDR value suggests sparse feature representations, increasing the difficulty of accurately classifying features into their gloss classes and this is often observed in gloss-free methods [2]. The formula is given as follows:

$$SDR(G_i) = \frac{D_{G_i}^{intra}}{avg.D_{G_i}^{inter}} = \frac{D(G_i)}{Mean_{j \neq i}(D(G_i, G_j))} \quad (1)$$

Where:

$$D(G_i, G_j) = \frac{1}{|G_i|}\frac{1}{|G_j|}\sum_{x \in G_i, y \in G_j} d(x, y)\,; \quad (2)$$

$$D(G_i) = \frac{1}{|G_i|}\frac{1}{|G_i-1|}\sum_{x,y\in G_i, x\neq y} d(x,y)\,; \quad (3)$$

Glosses G_i and G_j contain $|G_i|$ and $|G_j|$ instances respectively, and $d(x,y)$ denotes the Euclidean distance between the embeddings of instances x and y. The average SDR, $SDR = Mean(SDR(G_i))$, was calculated to evaluate the overall degree of representation density across all glosses.

The above formulas were coded to evaluate the compactness of feature representations, thereby determining the accuracy of the SLT model. **KMeans Clustering**, an unsupervised ML algorithm, was used to cluster video embeddings (i.e. feature vectors from sign videos) into their target words based on their similarity.

```
ALGORITHM SignDensityRatio(embeddings, num_clusters, target_words)
  INPUT:
    embeddings: n × d matrix of video embeddings # where n is the number of data points and d is the feature dimensions
    num_clusters: Number of target word clusters
    target_words: List of specific gloss words corresponding to clusters

  OUTPUT:
    sdr_values: Dictionary of SDR values for each key (target word)

  PROCEDURE:
  1. INITIALISE:
    labels ← KMeans(embeddings, num_clusters) # Cluster embeddings using KMeans
    clusters ← GroupByCluster(labels, embeddings) # Create clusters from labels

  2. DEFINE Function IntraGlossDistance(cluster)
    IF size(cluster) < 2 THEN
      RETURN 0 # Ignore clusters with fewer than 2 points
    END IF
    dist_matrix ← PairwiseDistances(cluster, cluster) # Calculate Intra-Gloss D(i,j)
    RETURN Mean(dist_matrix)
```

```
3. DEFINE Function InterGlossDistance(cluster1, cluster2)
   dist_matrix ← PairwiseDistances(cluster1, cluster2) # Calculate Inter-Gloss D(i,j)
   RETURN Mean(dist_matrix)

4. CALCULATE intra_distances FOR each cluster:
   FOR each cluster_id, cluster IN clusters DO
      intra_distances[cluster_id] ← IntraGlossDistance(cluster)
   END FOR

5. CALCULATE avg_inter_distances FOR each cluster:
   FOR each cluster_id, cluster IN clusters DO
      inter_dist_sum ← 0
      FOR each other_cluster_id, other_cluster IN clusters DO
         IF cluster_id ≠ other_cluster_id THEN
            inter_dist_sum ← inter_dist_sum + InterGlossDistance(cluster, other_cluster)
         END IF
      END FOR
      avg_inter_distances[cluster_id] ← inter_dist_sum / (num_clusters - 1)
   END FOR

6. COMPUTE SDR values FOR each cluster:
   FOR each cluster_id IN clusters DO
      sdr_values[target_words[cluster_id]] ← intra_distances[cluster_id] /
avg_inter_distances[cluster_id]
   END FOR

   RETURN sdr_values
END ALGORITHM
```

Figure 1: PseudoCode of the SDR when coded in Python

4.3 Machine Learning Methods

Classical ML methods were employed to address the supervised classification problem identified in this study. Methods such as the **Support Vector Machine** and **Random Forest** classifiers are effective tools for mapping video embeddings from the datasets to their corresponding target words. By evaluating and comparing the Precision, Recall, and F1-score of these methods, the study aims to identify which ML method will yield the best results on unseen data (i.e. continuous sign sentences) in the future, ultimately enabling meaningful and accurate SLT.

As the "Others" label was a significantly larger class (Refer to Appendix A for distribution of target words), **Synthetic Minority Over-sampling Technique (SMOTE)** [16] was employed when plotting the confusion matrix. SMOTE is a data augmentation technique that generates synthetic samples for the minority classes (i.e. all labels other than "Others"), preventing them from being overlooked, to ensure all classes are represented.

4.3.1 Support Vector Machine (SVM) Classifier

The SVM classifier was used to find the decision boundary, or hyperplane, that best separates data points corresponding to the different target words. 2 experiments were conducted to train the SVM using 2 different kernels – **'Linear'** and **'Radial Basis Function (RBF)'** respectively. Experiments (see results in Section 5) show that the 'RBF' kernel gives a better result than the 'Linear' kernel, with 0.0223 higher in Precision, 0.0214 higher in Recall, and 0.0223 higher in F1-score. This suggests that the relationship between the data points is non-linear and complex. As a result, the 'Linear' kernel is unable to classify the data using a single straight line, as it cannot effectively separate the data based on their respective target words.

4.3.2 Random Forest (RF) Classification

The RF algorithm is an ensemble approach that aggregates the outputs of multiple decision trees to produce a singular result. This ensures a higher degree of accuracy, controlling the issue of overfitting. Experiments showed that RF outperformed the 'RBF' SVM, with 0.0224, 0.0215, and 0.0225 higher in Precision, Recall, and F1-scores respectively. This higher performance could be attributed by the shape of the training data being (8257, 1024), corresponding to high-dimensional embeddings, and a large number of training data, conditions under which RF tends to excel.

5 Experimental Results

Figure 2: t-SNE Visualisation of PHOENIX-2014T Sign Features extracted using I3D

Table 1: SDR Values for I3D Model

Target Words	SDR Value (2 d.p.)
koennen	0.93
grad	0.89
gewitter	0.90
zwischen	0.92
nacht	0.92
freundlich	0.91
nordost	0.89
Average SDR Value: 0.91	

From the t-SNE visualisation plot in Figure 2, we observed that video embeddings from I3D, the gloss-free dataset, are very sparsely distributed. This was supported by our computed average SDR value of 0.91 as shown in Table 1, which we consider high based on Ye et al.'s work [2], which achieved an SDR value of 0.83 for his reproduced I3D gloss-free features and 0.66 for SMKD gloss-based features. This highlights the lack of distinct clustering in the I3D model, which makes it difficult for the model to carry out the video-to-gloss transcription.

Table 2: Overall evaluation of ML models for classifying sign videos into gloss

Model	Kernel	Precision	Recall	F1-score	Accuracy
SVM	Linear	0.8529	0.8646	0.8567	0.97
	RBF	0.9524	0.9534	0.9524	0.97
RF	-	**0.9748**	**0.9749**	**0.9749**	0.97

Table 2 shows that RF achieved the best results for Precision, Recall, and F1-score. Among the SVM models, 'RBF' performed better than the 'Linear' kernel, indicating the need for non-linear decision boundaries to classify video embeddings effectively.

Moreover, all the ML methods demonstrated exceptional classification results, with both the SVM and RF classifiers achieving weighted average Precision, Recall, F1 score, and Accuracy exceeding 0.95. (Weighted average was taken to account for the imbalanced dataset.) This proves that ML alone is indeed sufficient for the classification task, and consequently, video-to-gloss transcription.

6 Limitations and Future Work

To extract sign features from specific gloss words, we averaged the features from all frames in a video, as the reference paper [2] did not provide specific instructions on how this is done. This method may not be optimal, as it may combine different sign gestures. Ideally, we would calculate the duration of each sign feature and average the vectors over that duration to more accurately represent the gloss word, capturing its full temporal span. However, due to a lack of duration information, we used the simplified approach of averaging across all frames.

Additionally, due to time constraints, this paper focused primarily on the video-to-gloss pipeline and was unable to extend its research to include gloss-to-sentence SLT. As such, the sign videos are only classified by their corresponding target glosses, without producing full sentence translation for real-time SLT. Future work could also explore a broader range of datasets like the Singapore Sign Language, to enhance the research's applicability and impact in different local contexts.

7 Conclusion

While the t-SNE plot and SDR calculated for the I3D dataset did not exactly match results from Ye et al. [2], similar trends were observed. I3D, a gloss-free dataset, had embeddings sparsely distributed in the feature space, which corresponded to a high SDR value. However, findings

of this study have proven that classical ML models alone were able to effectively classify these embeddings into their corresponding gloss words, without the need for deep learning.

8 Acknowledgement

This project was supported by DSO National Laboratories as part of the Research@YDSP programme 2024. I would like to sincerely thank my research mentors, Ms Ip Hei Man, Dr Woo Chin Jian, and Dr Shen Bingquan for their invaluable guidance and support, and for the opportunity to embark on this research project.

References

[1] Sign language. National Geographic. (n.d.). https://education.nationalgeographic.org/resource/sign-language/.

[2] Ye, J., Wang, X., Jiao, W., Liang, J., & Xiong, H. (2024, October 28). *Improving gloss-free sign language translation by reducing representation density.* arXiv.org. https://arxiv.org/abs/2405.14312.

[3] Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). *Neural sign language translation.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7784–7793.

[4] Hao, A., Min, Y., and Chen, X. (2021). *Self-mutual distillation learning for continuous sign language recognition.* In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11303– 11312.

[5] Joao Carreira and Andrew Zisserman. (2018). *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[6] Zhou, B., Chen, Z., Clapés, A., Wan, J., Liang, Y., Escalera, S., Lei, Z., and Zhang, D. (2023). *Gloss-free sign language translation: Improving from visual-language pretraining. I*n Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 20871–20881.

[7] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2020). *A survey on contrastive self-supervised learning.* Technologies, 9(1):2.

[8] Yin, A., Zhong, T., Tang, L., Jin, W., Jin, T., and Zhao, Z. (2023). Gloss attention for gloss-free sign language translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2551–2562.

[9] Moryossef, A., Müller, M., Göhring, A., Jiang, Z., Goldberg, Y., & Ebling, S. (2023, May 28). *An open-source gloss-based baseline for spoken to signed Language Translation.* arXiv.org. https://arxiv.org/abs/2305.17714.

[10] Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., & Gao, J. (2019, December 4). *Unified vision-language pre-training for image captioning and VQA*. arXiv.org. https://arxiv.org/abs/1909.11059.

[11] Lin, K., Wang, X., Zhu, L., Sun, K., Zhang, B., & Yang, Y. (2023, May 27). *Gloss-free end-to-end sign language translation*. arXiv.org. https://arxiv.org/abs/2305.12876.

[12] Van der Maaten, L. and Hinton, G. (2008). *Visualizing data using t-sne.* Journal of Machine Learning Research, 9(11).

[13] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. *The Kinetics human action video dataset.* arXiv.org. https://arxiv.org/pdf/1705.06950.

[14] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li. *ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition*. In CVPR ChaLearn Looking at People Workshop, 2016.

[15] Sarhan N, Frintrop S. *Transfer learning for videos: from action recognition to sign language recognition.* In the 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020: 1811-1815.

[16] Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). *SMOTE: Synthetic minority over-sampling technique.* arXiv.org. https://arxiv.org/pdf/1106.1813.

Appendix

A Distribution of Target Words

Figure 3 shows the distribution of each target word. Results show that the vast majority of the samples fall under the "Others" class, which results in an imbalanced dataset. This was the reason that led to SMOTE being applied before training the data using the classical ML models, and for Weighted Average to be taken as the overall evaluation metric.

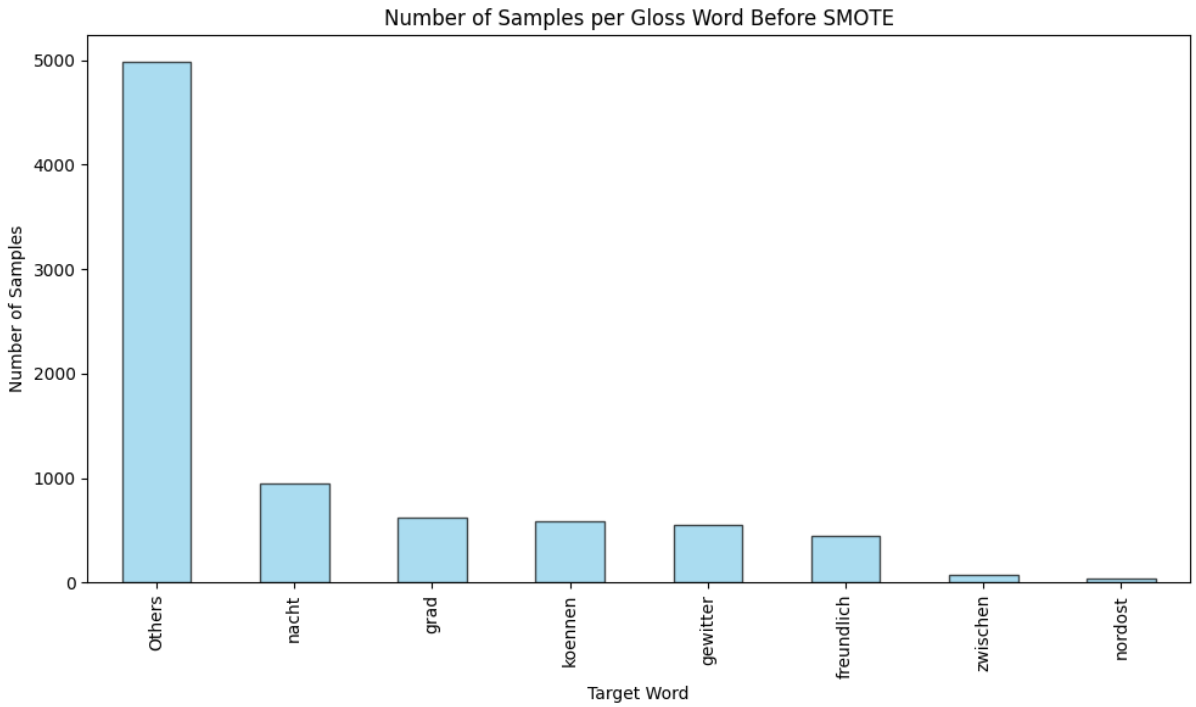


Figure 3: Bar graph of the number of samples per gloss word before SMOTE was applied

B Confusion Matrix and Classification Reports for Machine Learning training

Table 3: Comparison of confusion matrices for different SVM kernels

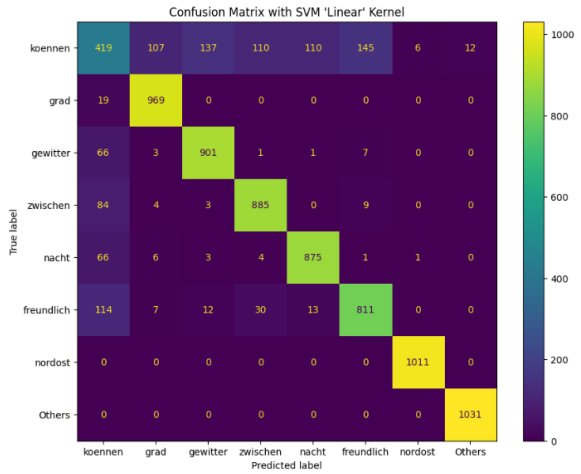
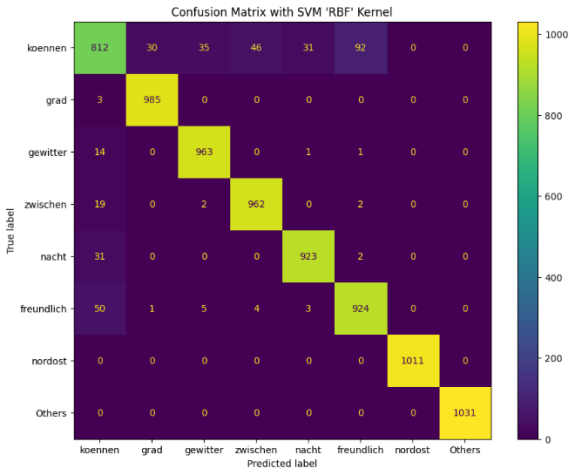
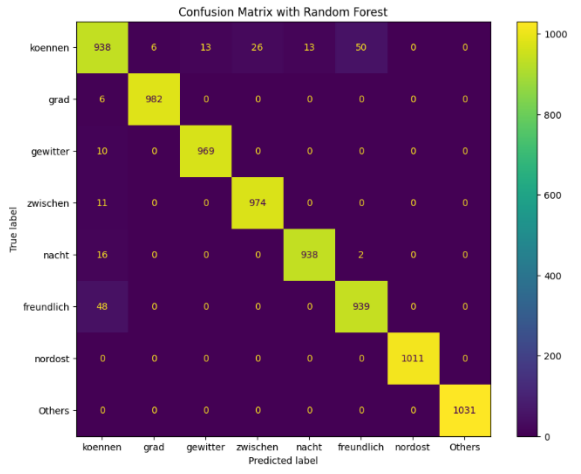
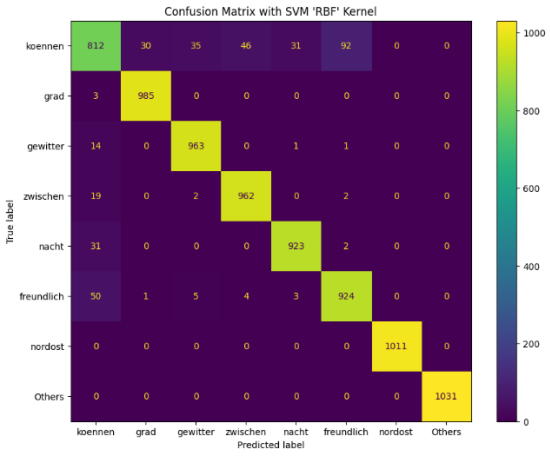
SVM (kernel: 'Linear')	SVM (kernel: 'RBF')
	
Conclusion: 'RBF' kernel performs better than 'Linear' kernel using SVM.	

Table 4: Comparison of confusion matrices for RF and SVM

RF (100 trees)	SVM (kernel: 'RBF')
	
Conclusion: RF performs better than SVM using the 'RBF' kernel.	

C Confusion Matrix and Classification Report for Cross-Validation

Figure 4 shows the confusion matrix and classification report produced after cross-validation is applied on the I3D model trained using RF. As RF was identified earlier as the best-performing classical ML model for training I3D, this cross-validation serves as a precautionary measure to ensure that over-fitting has not occurred.

Over-fitting would result in a high performance on the training data, but a poor performance on unseen test data. However, from the results, it can be confirmed that the I3D model does not suffer from overfitting after it was trained with RF. Rather, it was still able to learn meaningful patterns from the video embeddings and classify them into their corresponding gloss words effectively.

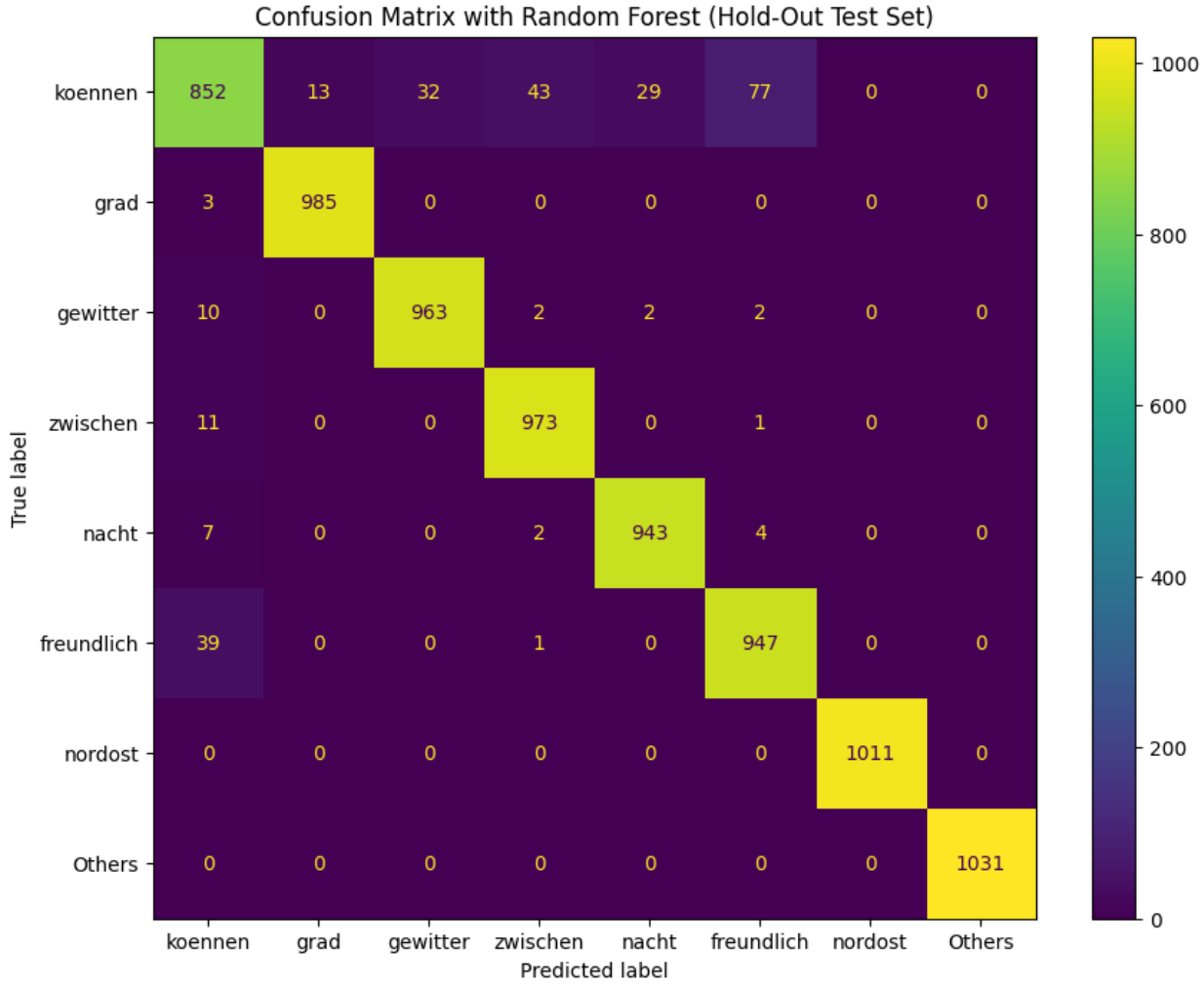


Figure 4: Confusion matrix for cross-validation with RF

Table 5: Classification report for cross validation (hold-out test set)

	Precision	Recall	F1-Score	Support
koennen	0.92	0.81	0.87	1046
grad	0.99	1.00	0.99	988
gewitter	0.97	0.98	0.98	979
zwischen	0.95	0.99	0.97	985
nacht	0.97	0.99	0.98	956
freundlich	0.92	0.96	0.94	987
nordost	1.00	1.00	1.00	1011
Others	1.00	1.00	1.00	1031

Accuracy: 0.97 (Support: 7983)
Weighted Average Precision (Test Set): 0.9648
Weighted Average Recall (Test Set): 0.9652
Weighted Average F1 Score (Test Set): 0.9644

Table 6: Metric evaluation for cross-validation

Metric Evaluation	Mean	SD
Precision	0.9746	0.0008
Recall	0.9748	0.0008
F1	0.9746	0.0008