Domain-sum Feature Transformation Applied to Sign Language Feature Extraction

Matheus Silva de Lima

11/07/2024

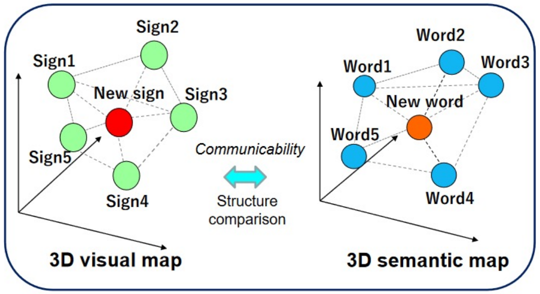
Agenda

- Introduction
- Background
- Proposed Research
- Results-so-far
- Next-up

Introduction

- Sign languages are **visual-based languages**

- On our first work, we proposed a metric to measure similarity between semantic and visual modes




Home > Conference proceedings

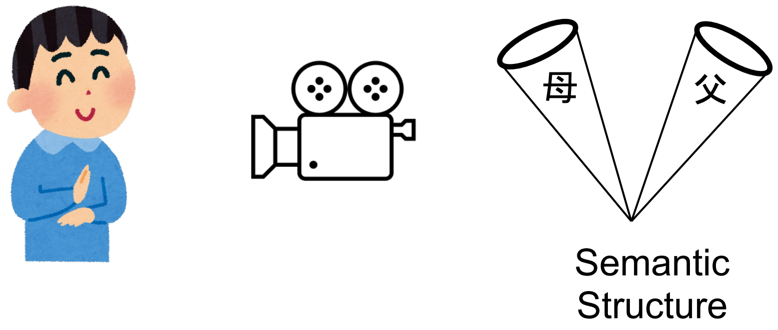
Frontiers of Computer Vision

30th International Workshop, IW-FCV 2024, Tokyo, Japan, February 19–21, 2024, Revised Selected Papers

Conference proceedings | © 2024

Introduction

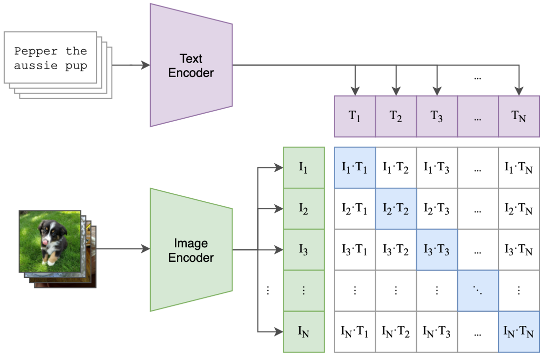
- This time we are tackling feature-generation process.



- How to generate features that better incorporate semantic meaning?

Introduction

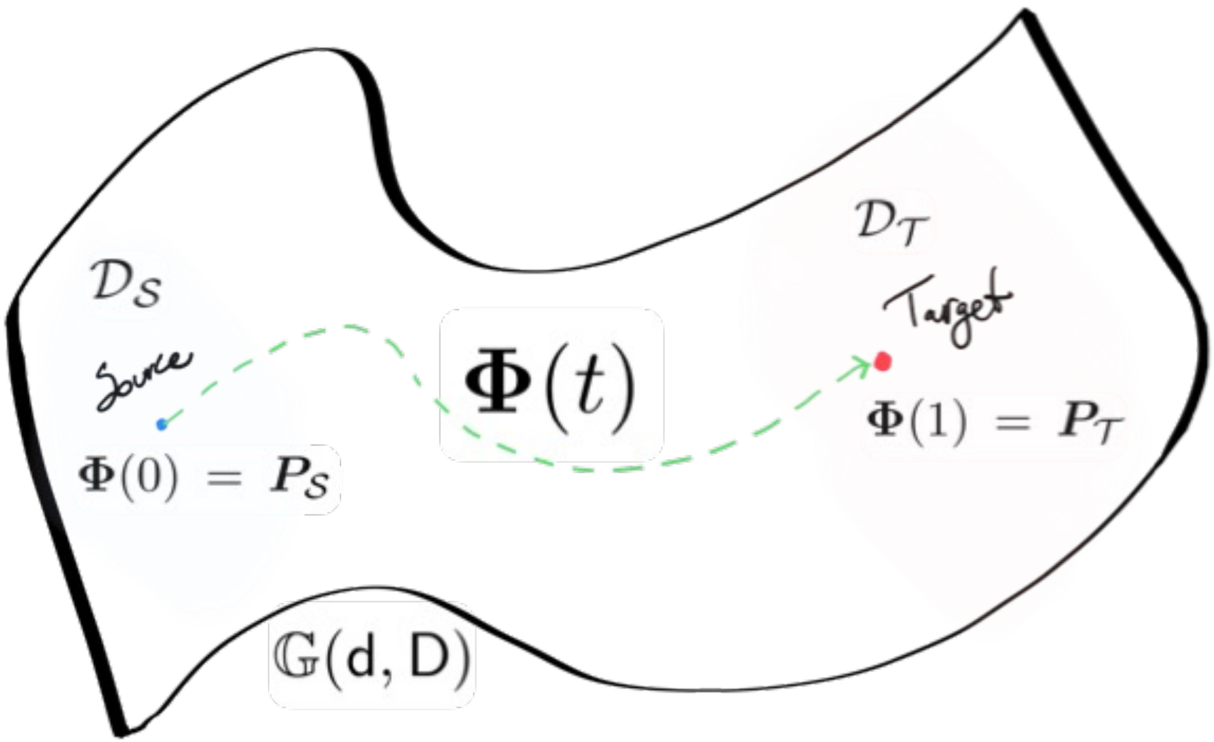
- Learning Transferable Visual Models From Natural Language Supervision (2021)



- Many models tackle this problem: ViFi-CLIP, InternVideo, etc…

- We are interested in investigating an approach called **Domain-sum Feature Transformation**

Background: Geodesic flow kernel for unsupervised domain adaptation

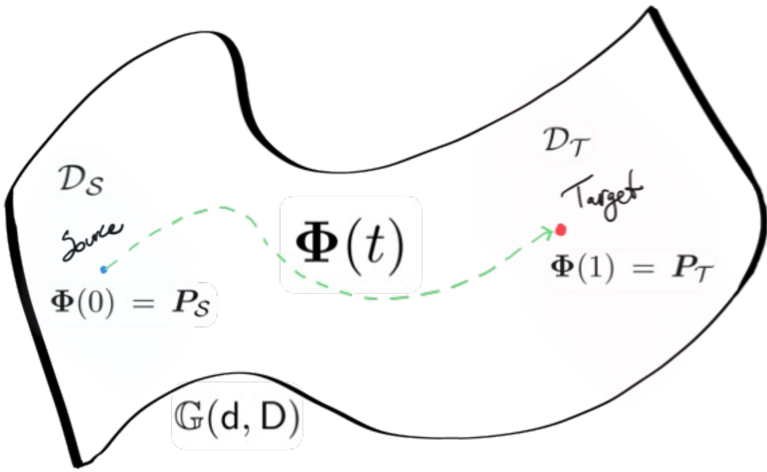


Background: Geodesic flow kernel for unsupervised domain adaptation

- Constructs the **geodesic flow**, which parametrizes how source domain smoothly transitions to target domain

$$\boldsymbol{\Phi}(t) = \boldsymbol{P}_{\mathcal{S}} \boldsymbol{U}_1 \boldsymbol{\Gamma}(t) - \boldsymbol{R}_{\mathcal{S}} \boldsymbol{U}_2 \boldsymbol{\Sigma}(t),$$

$$\boldsymbol{P}_{\mathcal{S}}^{\mathrm{T}} \boldsymbol{P}_{\mathcal{T}} = \boldsymbol{U}_1 \boldsymbol{\Gamma} \boldsymbol{V}^{\mathrm{T}}, \quad \boldsymbol{R}_{\mathcal{S}}^{\mathrm{T}} \boldsymbol{P}_{\mathcal{T}} = -\boldsymbol{U}_2 \boldsymbol{\Sigma} \boldsymbol{V}^{\mathrm{T}}.$$



- By projecting features vectores x_i into $\phi(t)$ for *all* continuous values of $t \in [0, 1]$, we get an infinite-size feature vector $z_i^\infty = \phi(t)^T x_i$.

- The inner-product $\langle z_i^\infty, z_j^\infty \rangle$ of two feature vectors x_i, x_j is the **geodesic flow kernel (GFK)**

$$\langle \boldsymbol{z}_i^\infty, \boldsymbol{z}_j^\infty \rangle = \int_0^1 (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_j)\, dt = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{G} \boldsymbol{x}_j$$

Background: Domain-Sum Feature Transformation For Multi-Target Domain Adaptation

- **GFK** is harder to incorporate in end-to-end learning.

- Kobayashi et al. [2] proposes that GFK integral can be approximate by the sum of the geodesic flow's extremes.

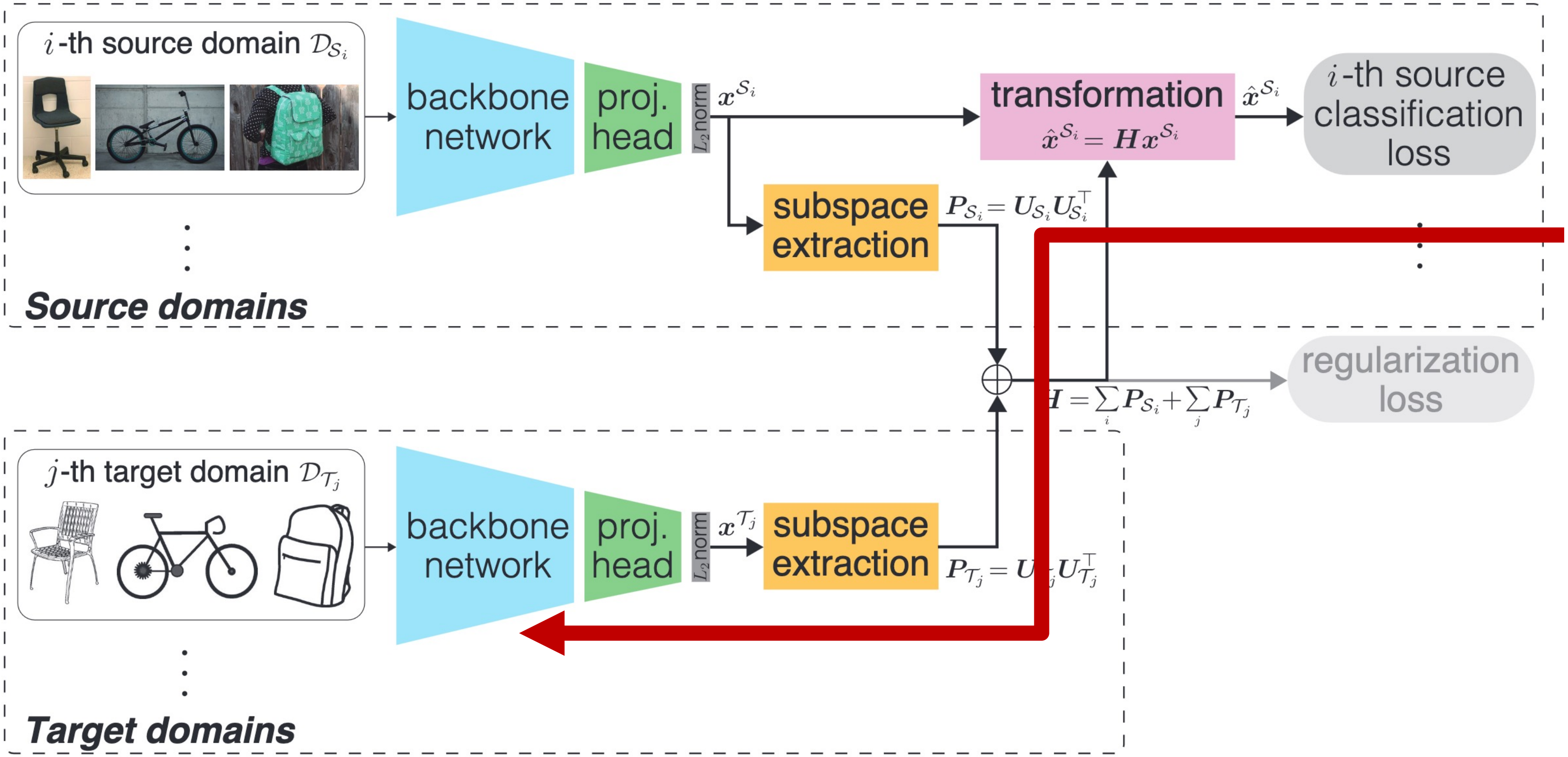
$$G = \int_0^1 \phi(t)\phi(t)^T dt \quad \Rightarrow \quad H = \sum_{t\in\{0,1\}} \phi(t)\phi(t)^T$$

- They propose a feature transformation using $\widehat{H} = H^2$ for the domain adaptation, as

$$x_i^T \widehat{H} x_j = (Hx_i)^T Hx_j \ \rightarrow \ \hat{x} = Hx$$

- This is the so-called **domain-sum feature transformation**.

Background: Domain-Sum Feature Transformation For Multi-Target Domain Adaptation



Background: Domain-Sum Feature Transformation For Multi-Target Domain Adaptation

- Problem: differentiability of SVD operator

SVD backward isn't correctly defined when multiple 0 eigenvalues #49886

Clos [feature request] A rank-revealing SVD for better stability in backward. #69532

Open `(svd|pca)_lowrank`: backward is unstable when for a matrix `A`, the parameter `q` is set to a value `q > rank(A)`. #69531

Open nikit Consistent treatment of non-differentiability in linear algebra operations #57272

Open **lezcano** opened this issue on Apr 29, 2021 · 2 comments

Lincon 10h15
Oi Mateus, eu to de volta ao trabalho hoje.
Sobre o stable/robust SVD: Parece q vc resolveu sua duvida ne.
Mas assim, vou explicar qual eh a questao do SVD, pq ele precisa ser "robusto".
O fato eh que o SVD nao tem uma derivada definida em todo o espectro.
O PDF em anexo demonstra a derivada do SVD, eh bem complicado mas mando pra referencia.
Existe uma parte onde eh necessario o calculo do reciproco da diferença entre dois autovalores,
(a equação entre 11 e 12, no texto, que comeca com F_ij). Se os dois autovalores s_i e s_j forem iguais, a diferenca é zero e o reciproco é indefinido, inviabilizando toda a cadeia de cálculo.
Faz sentido? Espero que isso ajude a entender porque o SVD é instavel na região de autovalores próximos.
E isso não é uma fronteira nítida, é uma função contínua que cresce exponencialmente, então até mesmo valores diferentes, mas próximos o suficiente para explodir o gradiente bastam para inviabilizar o SVD no forward de uma rede neural. Na minha experiência isso se manifesta como erros de convergencia na função torch.svd e/ou torch.eig.

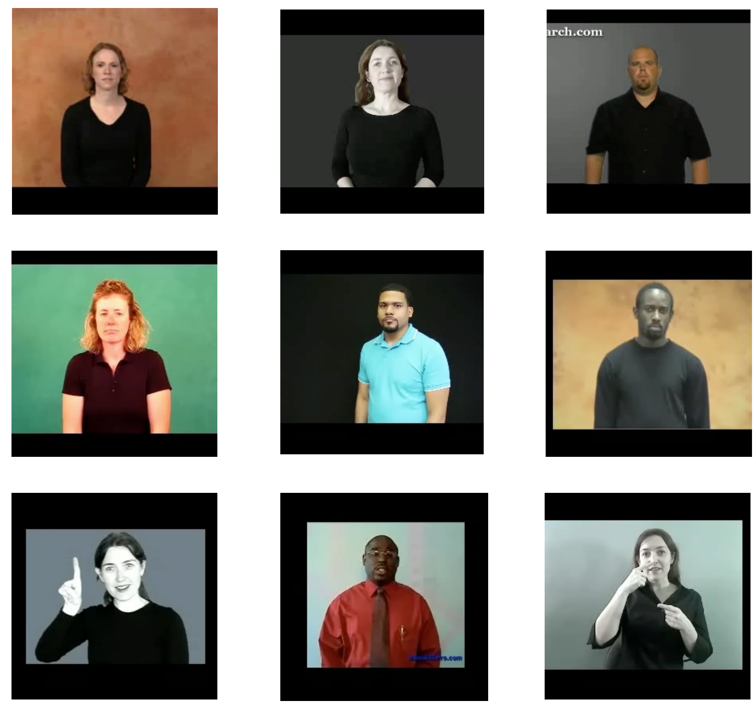
- The differential of SVD operator depends on a matrix $\widetilde{K}$ inversely proportional to the difference of eigenvalues.

$$\widetilde{K}_{ij} = \begin{cases} 1/(\lambda_i - \lambda_j), & i \neq j \\ 0, & i = j \end{cases}$$

- To solve this, some methods have been proposed using Taylor approximation [3] and Padé approximations [4]. Kobayashi et al. used Taylor approximations.

Background: Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison

- A new dataset: WLASL2000
    - 2000 Glosses (labels)
    - 21.083 Videos



- Classified this dataset using 4 different architectures

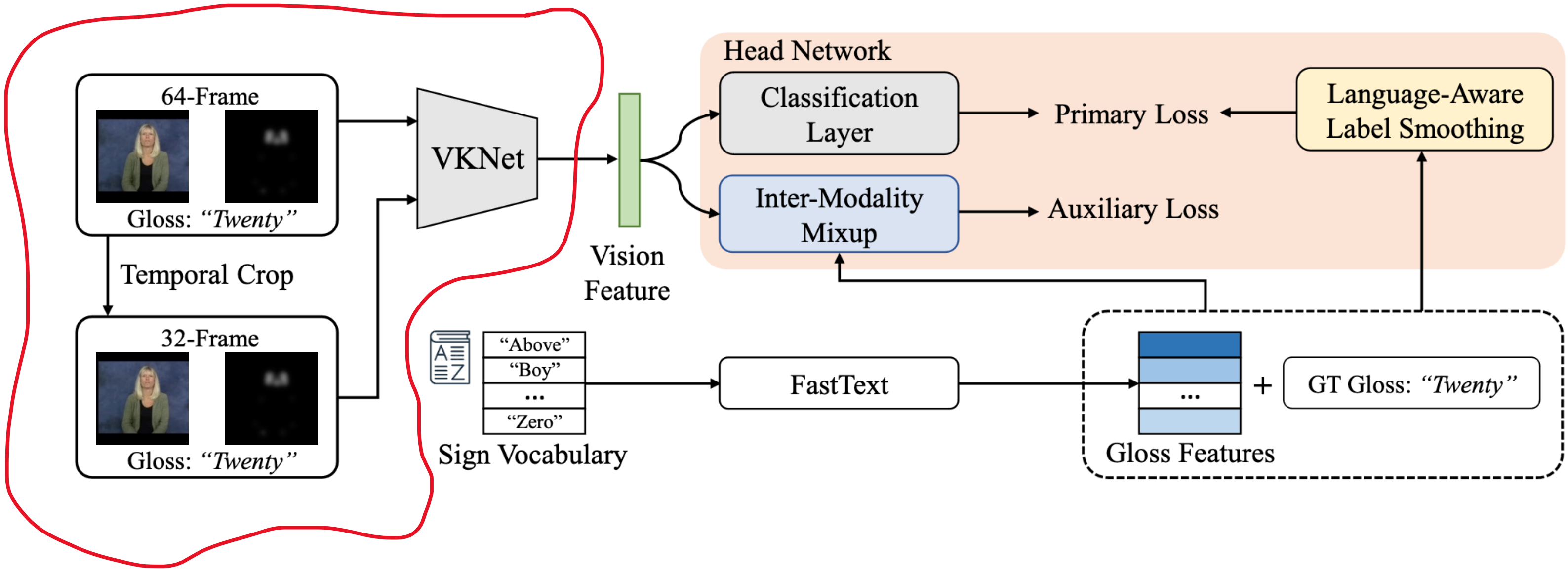
Method	WLASL2000		
	top-1	top-5	top-10
Pose-GRU	22.54	49.81	61.38
Pose-TGCN	23.65	51.75	62.24
VGG-GRU	8.44	23.58	32.58
I3D	**32.48**	**57.31**	**66.31**

Background: Natural Language-Assisted Sign Language Recognition

- NLA-SLR: (current) state-of-the-art model on WLASL2000

Method	WLASL2000			
	Per-instance		Per-class	
	Top-1	Top-5	Top-1	Top-5
NLA-SLR (Ours)	61.05	91.45	58.05	90.70
NLA-SLR (Ours, 3-crop)	**61.26**	**91.77**	**58.31**	90.91

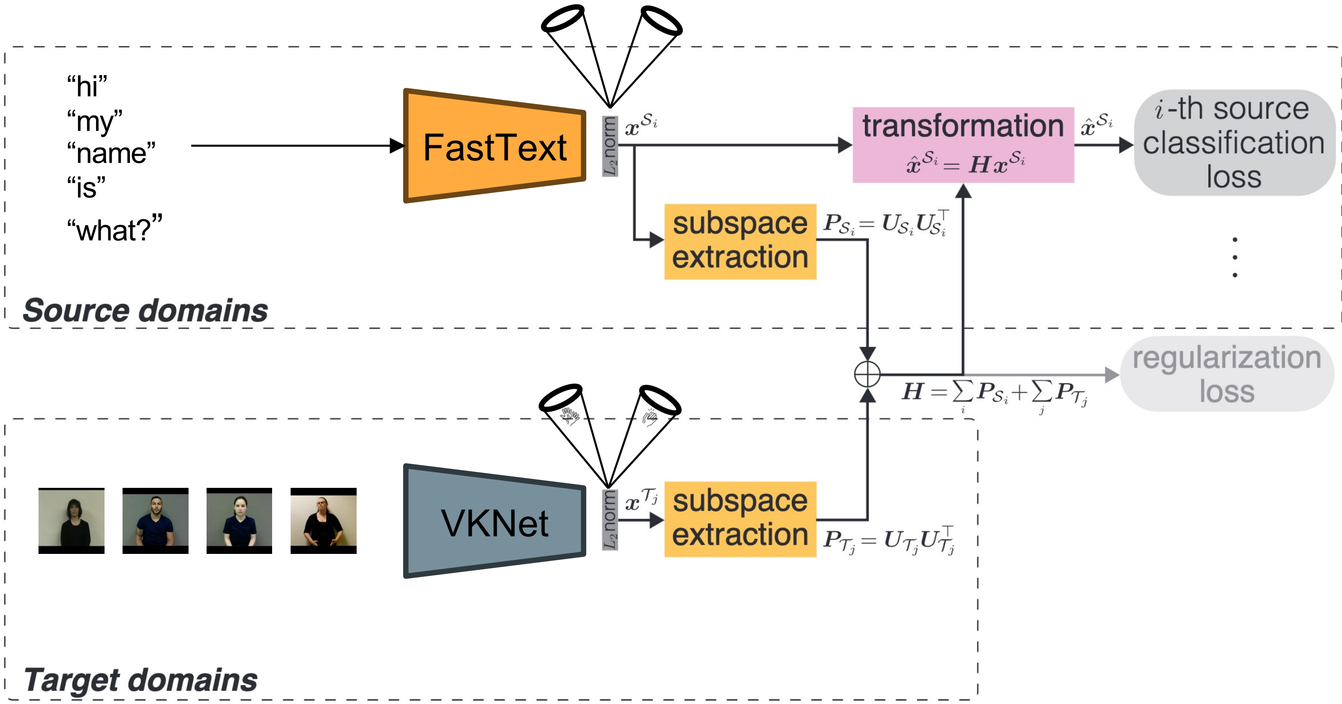
- Classification head incorporates language knowledge using label-smoothing and mix-up.



Proposed Research

- Domain-sum Feature Tranformation has been shown to be effective in bridging the gap between same-modal datasets:
  - Ex: **photos** of cars and **drawings** of cars
- Could it be effective for datasets in different modes?
  - Ex: **Visual**-language and **Textual**-language.
- Can we generate a richer semantic feature-space from the videos by bridging the gap between visual and textual representation?

Proposed Research

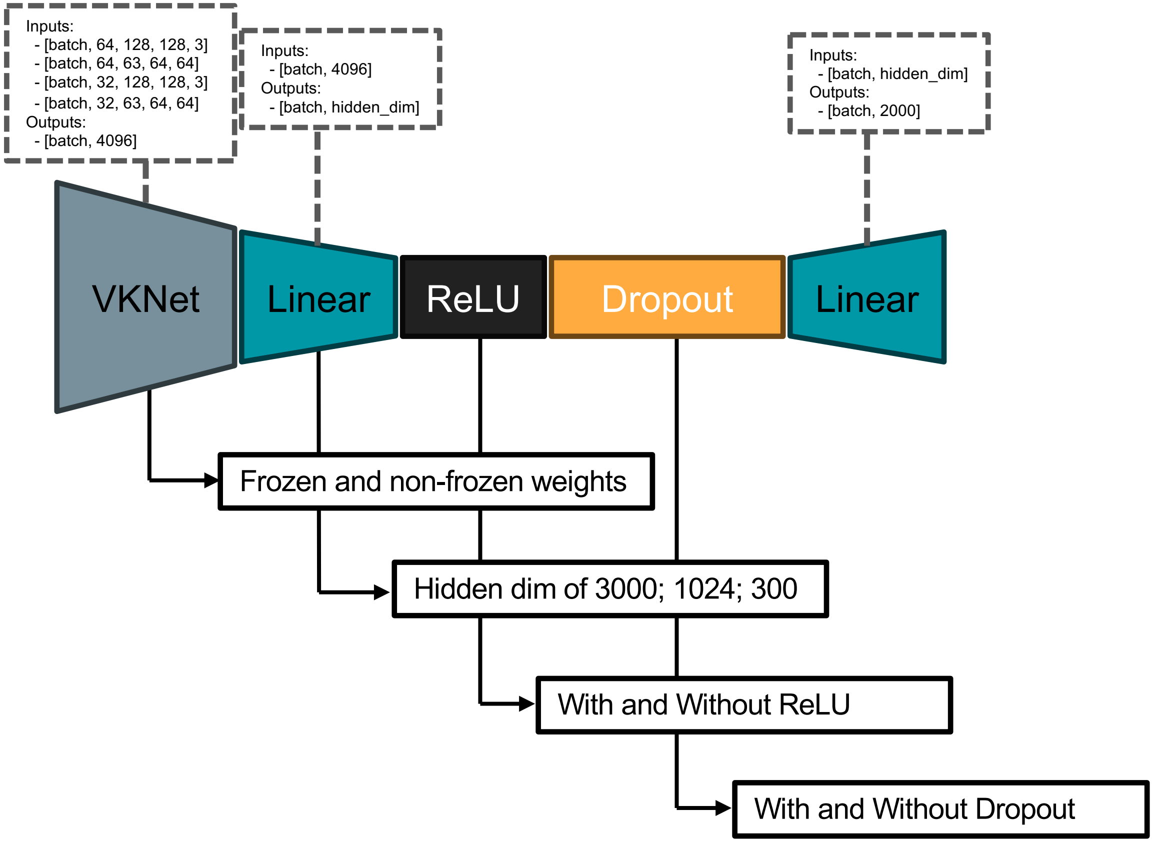


Proposed Research

- Does the method converge?
- Does it improve classification accuracy?
- Does it improves the communicability between visual-semantic modes?

Results-so-far

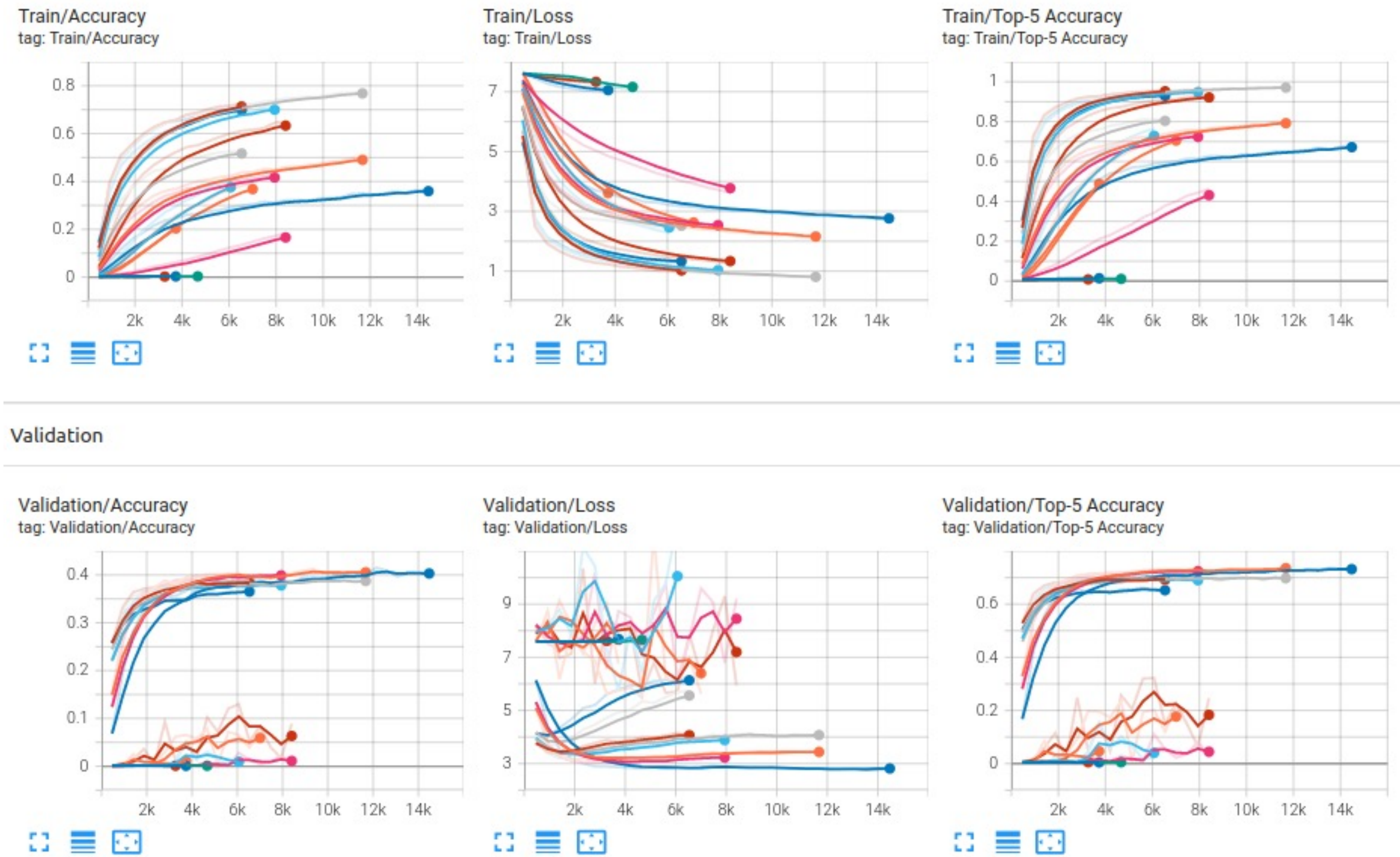
- Trained VKNet on WLASL2000 from pre-trained weights, removing language head.




```
batch_size: 10
min_number_of_frames: 25
 criterion:
   cross_entropy_loss:
     reduction: mean
 optimizer:
   adam:
     lr: 0.001
     weight_decay: 0.001
     betas:
       - 0.9
       - 0.998
 lr_scheduler:
   cosine_annealing_lr:
     eta_min: 0
     T_max: 100
```

Results-so-far

- VKNet trainning stats



Next-up

- Test training using video augmentation.
- Calculate communicability from current model (torchmetrics).
- Implement domain-sum architecture. Especially, implement SVD backwards pass (torch.autograd).
- Vizualize and compare metrics.

CVLAB 筑波大学 University of Tsukuba

Thank you

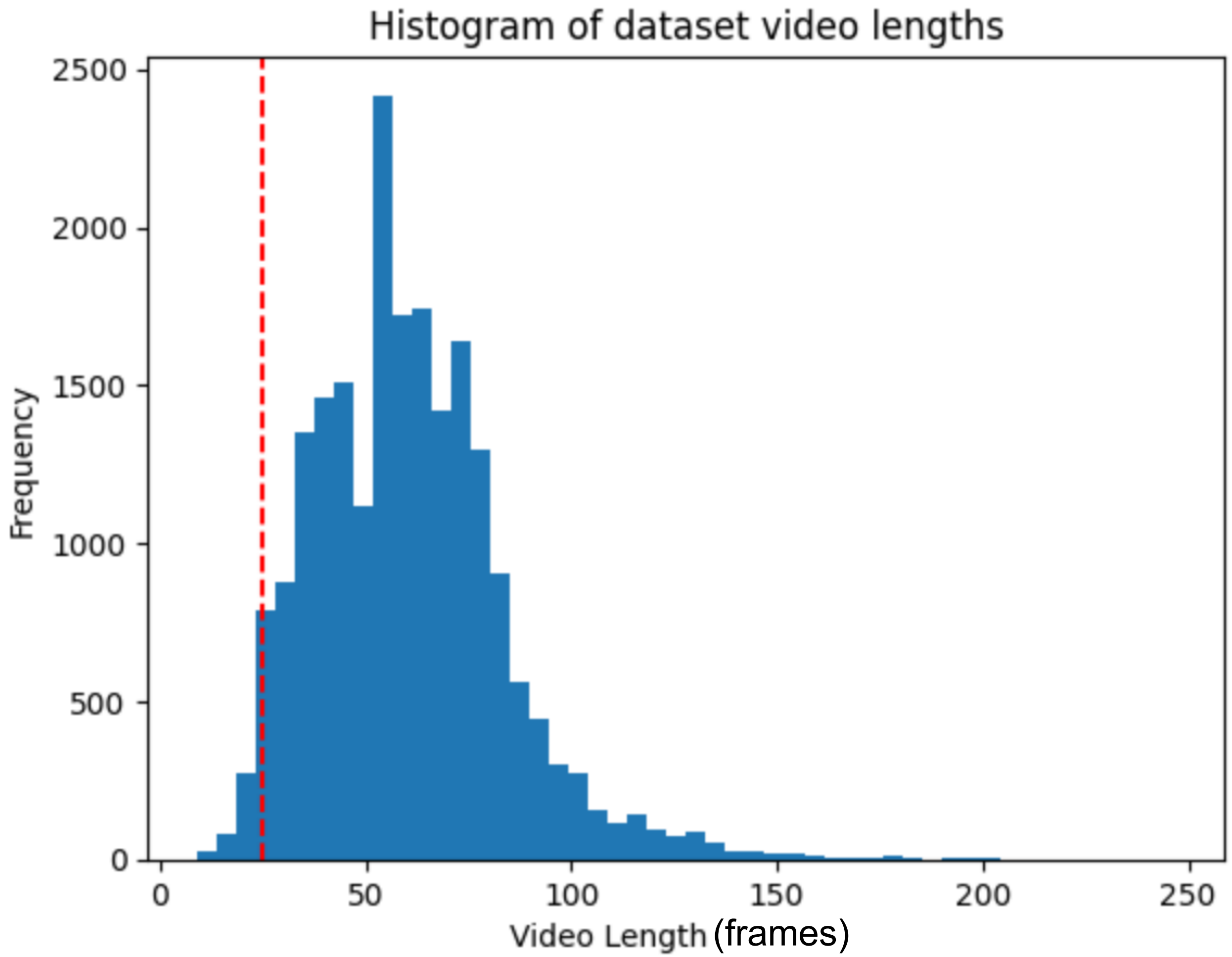
CVLAB 筑波大学 University of Tsukuba

References

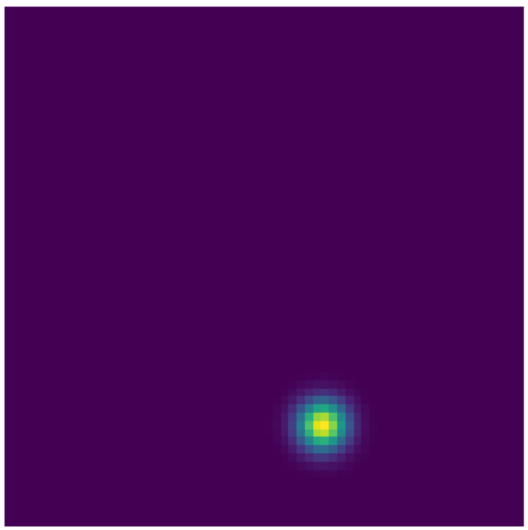
1. Geodesic flow kernel for unsupervised domain adaptation
2. Domain-Sum Feature Transformation For Multi-Target Domain Adaptation
3. Robust Differentiable SVD
4. Why Approximate Matrix Square Root Outperforms Accurate SVD in Global Covariance Pooling
5. WLASL: A large-scale dataset for Word-Level American Sign Language
6. Natural Language-Assisted Sign Language Recognition
7. Differentiating the Singular Value Decomposition

Extras

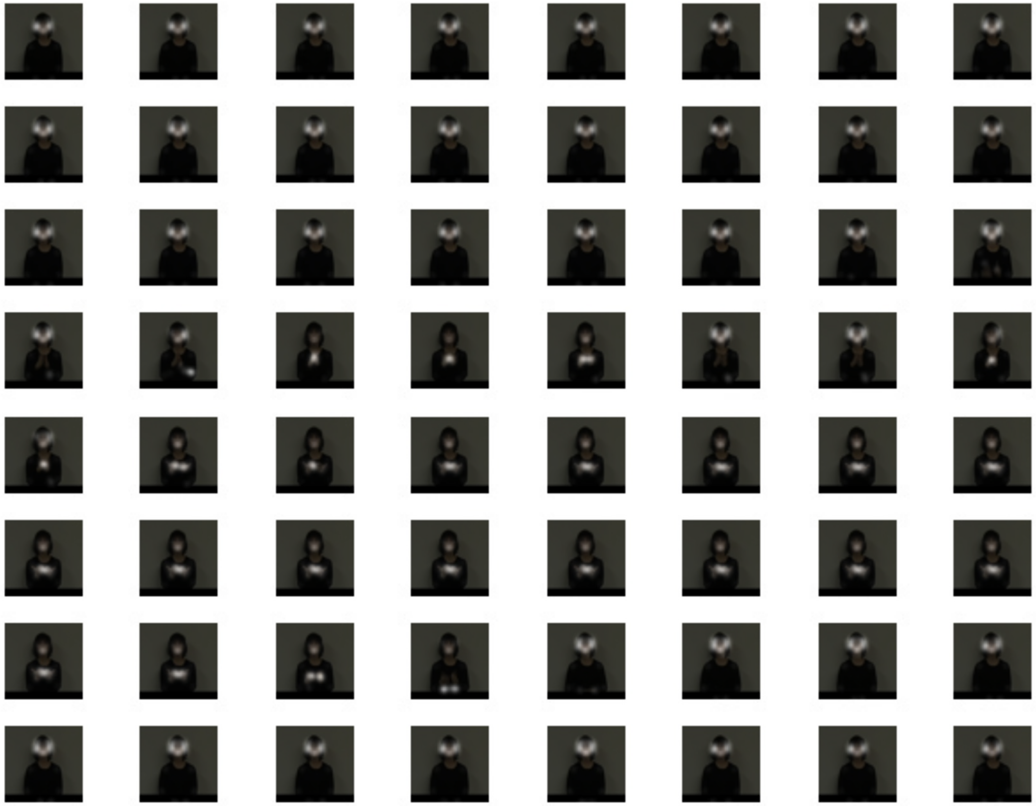
Proportion of numbers greater than n=25: 0.9778494521652517 467 count=20616



Extras



Example of keypoint heatmap



Mean heatmaps overlayed on video

Extras

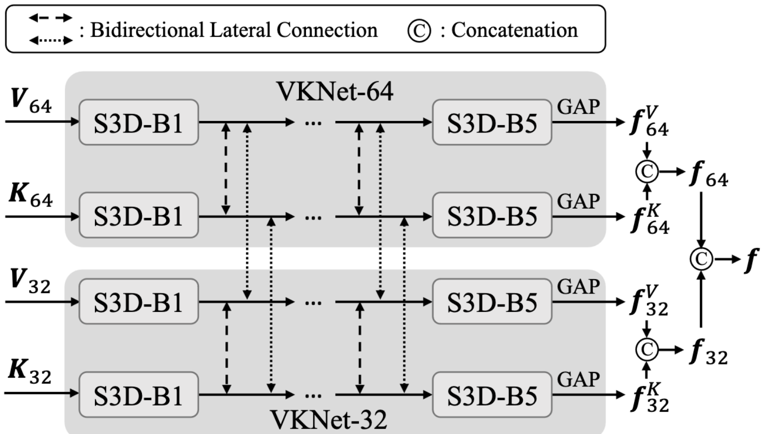


Figure 4. Our VKNet consists of two sub-networks, VKNet-64 and VKNet-32, which take video-keypoint pairs with different temporal receptive fields as inputs and output a set of vision features via global average pooling (GAP) layers. Within the VKNet, bidirectional lateral connections [9] are applied to the outputs of the first four S3D blocks (B1-B4) for video-video, keypoint-keypoint, and video-keypoint information exchange.

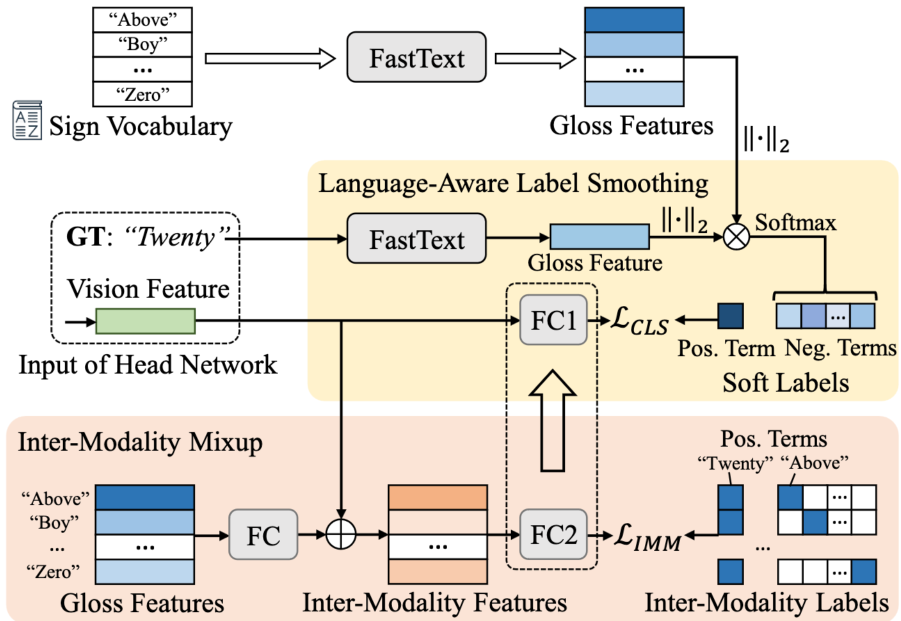


Figure 5. The architecture of our head network. Language-aware label smoothing generates soft labels whose smoothing weights are the normalized semantic similarities between the ground truth and remaining glosses within the sign vocabulary. Inter-modality mixup generates inter-modality features and the corresponding labels to maximize the signs' separability in a latent space. Integration between FC1 and FC2 can further boost SLR performance.

$$\boldsymbol{y}[i] = \begin{cases} 1-\epsilon & \text{if } i=b, \\ \epsilon \cdot \frac{\exp(\boldsymbol{s}[i]/\tau)}{\sum_{i=1,i\neq b}^{N}\exp(\boldsymbol{s}[i]/\tau)} & \text{otherwise,} \end{cases}$$

Extras

Algorithm 1 Subspace extraction module

Input: $\boldsymbol{X} \in \mathbb{R}^{d \times B}$: features on a mini-batch of size B,
r: subspace rank,
$\mathtt{M} \in \mathbb{R}^{d \times Q}$: memory bank to store samples,

Output: $\boldsymbol{P} \in \mathbb{R}^{d \times d}$: subspace projection matrix

1: $[\boldsymbol{X}, \mathtt{M}] = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$: SVD [8]
2: $\boldsymbol{P} = \boldsymbol{U}_{:r}\boldsymbol{U}_{:r}^\top$: extract the first r-rank basis vectors
3: Enqueue $\boldsymbol{X}$ to $\mathtt{M}$ and dequeue old samples from $\mathtt{M}$
4: **return** $\boldsymbol{P}$

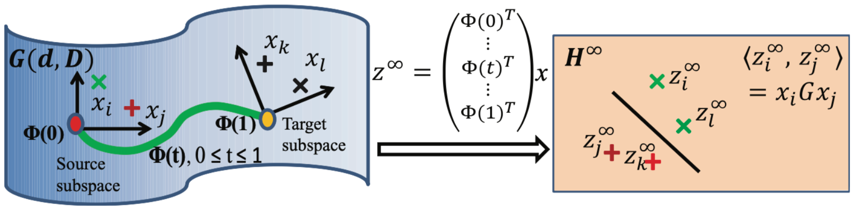


Figure 1. Main idea of our geodesic flow kernel-based approach for domain adaptation (Best viewed in color). We embed source and target datasets in a Grassmann manifold. We then construct a geodesic flow between the two points and integrate an infinite number of subspaces along the flow $\boldsymbol{\Phi}(t)$. Concretely, raw features are projected into these subspaces to form an infinite-dimensional feature vector $\boldsymbol{z}^\infty \in \mathcal{H}^\infty$. Inner products between these feature vectors define a kernel function that can be computed over the original feature space in closed-form. The kernel encapsulates incremental changes between subspaces that underly the difference and commonness between the two domains. The learning algorithms thus use this kernel to derive low-dimensional representations that are invariant to the domains.