Oral Presentation

HASPER: An Image Repository for Hand Shadow Puppet Recognition

Presented by

Syed Rifat Raiyan

Co-authors

Zibran Zarif Amio, Sabbir Ahmed

Department of Computer Science and Engineering

Islamic University of Technology

ICCV OCT 19-23, 2025 HONOLULU HAWAII

Workshop on Cultural Continuity of Artists

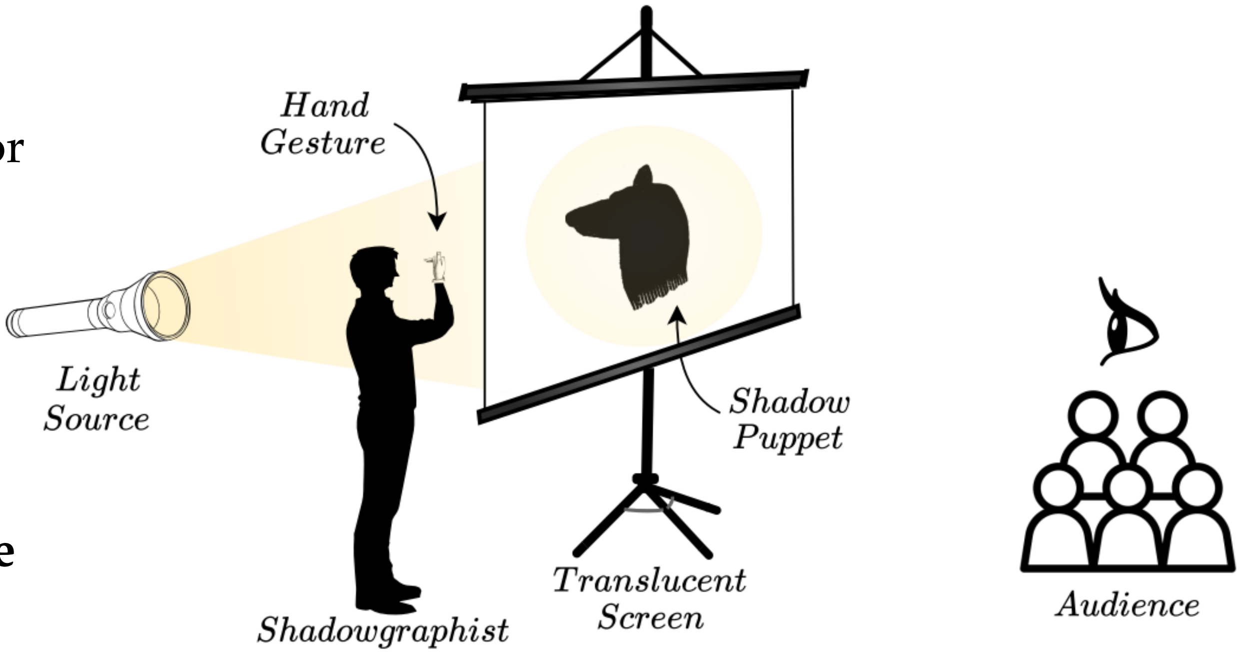
Introduction

What is Hand Shadow Puppetry?

Definition:
Hand Shadow Puppetry, also known as *shadowgraphy* or *ombromanie* is the art of performing a story or show using images made through the construction and manipulation of shadow figures or silhouettes, using one's hands, body, or props [1].

How does it work?
The puppeteer places his **hands between a light source and a translucent screen** to create shadows or silhouettes that resemble different **animals**.

- Also known as *cinema in silhouette*
- On the verge of extinction—UNESCO designated shadow puppetry an **endangered** Intangible Cultural Heritage [2] in 2011 (hence, needs **preservation**!)



(a) A generic hand shadow puppetry setup.

(b) Rabbit **(c)** Bird **(d)** Dog

Fig: Ombromanie in a nutshell.

Motivation

Our Inspiration to Pursue this Topic

- **Novelty factor:** To the best of our knowledge, **no** explicitly vision-related work or dataset exists on this topic of hand shadow puppet classification.
  - ➢ Some of the closely related works will be mentioned in a bit…
- **Gap:** Frontier image generator models are **very bad** at ombromanie.
- **Utility:**
  - ✓ **Tool for teaching** performance art
  - ✓ **Recreational app** for kids
  - ✓ Enabling the development of sophisticated algorithms for automatic **recognition**, **classification**, or even **generation** of ombromanie performances
- **Nostalgia** — incentivized by childhood memories during the load-shedding days.


ByteDance Seedream-4


xAI Grok 4


Google Imagen-4 Fast


Stable Diffusion 3.5


Google Gemini 2.5 Pro


Tencent Hunyuan 3

Related Work and Dataset

Research Literature on Shadow Puppetry

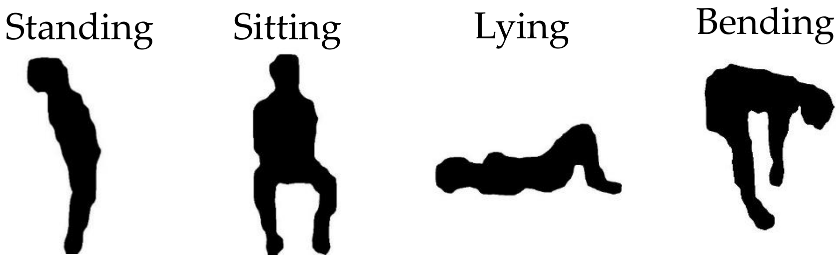
As mentioned before, we **haven't found any prominent work** on hand shadow puppet image classification.

Prominent Works: (closely related topics)

- In **Robotics,**
  - (Huang *et al.*)[3] — introduced a framework that enables **robotic arms** to **perform hand shadow puppetry** by matching shape correspondences of input image.
- In **Human-Computer Interaction,**
  - (Zhang *et al.*)[4] — worked on **emulating** the movements and body **gestures of a performer** on **Chinese shadow puppets** using Kinect sensor.
  - (Carr *et al.*)[5] — built a real-time **Indonesian shadow puppet** storytelling application using the Microsoft Kinect sensor capable of **mimicking full-body actions** of user.
- In **Computer Graphics,**
  - (Huang *et al.*)[6] — generated **3D models** of animals from shadow puppet images.

Dataset:

- Human Posture Silhouettes [7] — 4,800 binary images of silhouettes used for **human posture recognition**.



Our Work

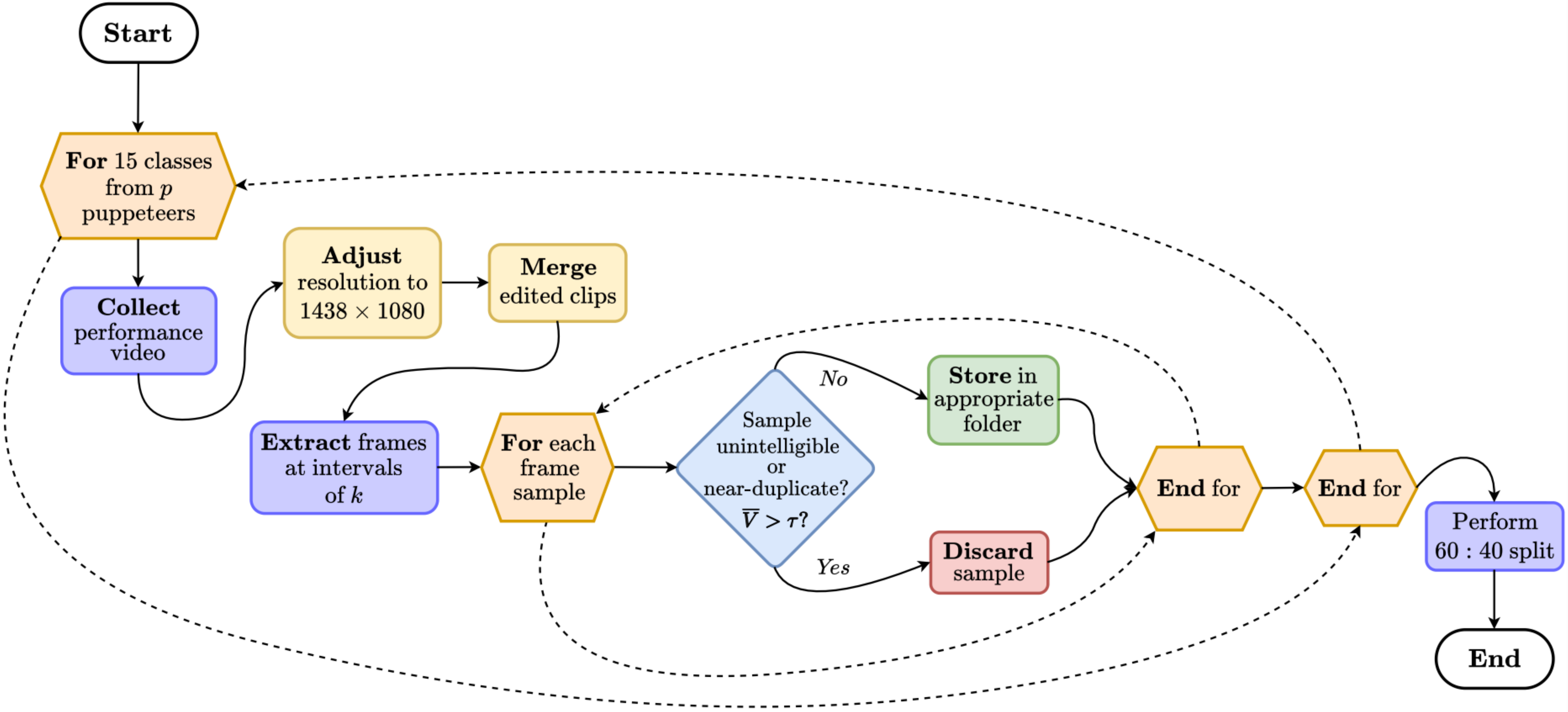
What the project entails

- **Data Collection** — Gathered a total of **15,000 images** of hand shadow puppets.
  - ✓ From **68** professional hand shadow puppeteer clips and **90** amateur clips.
  - ✓ Across **15** classes.

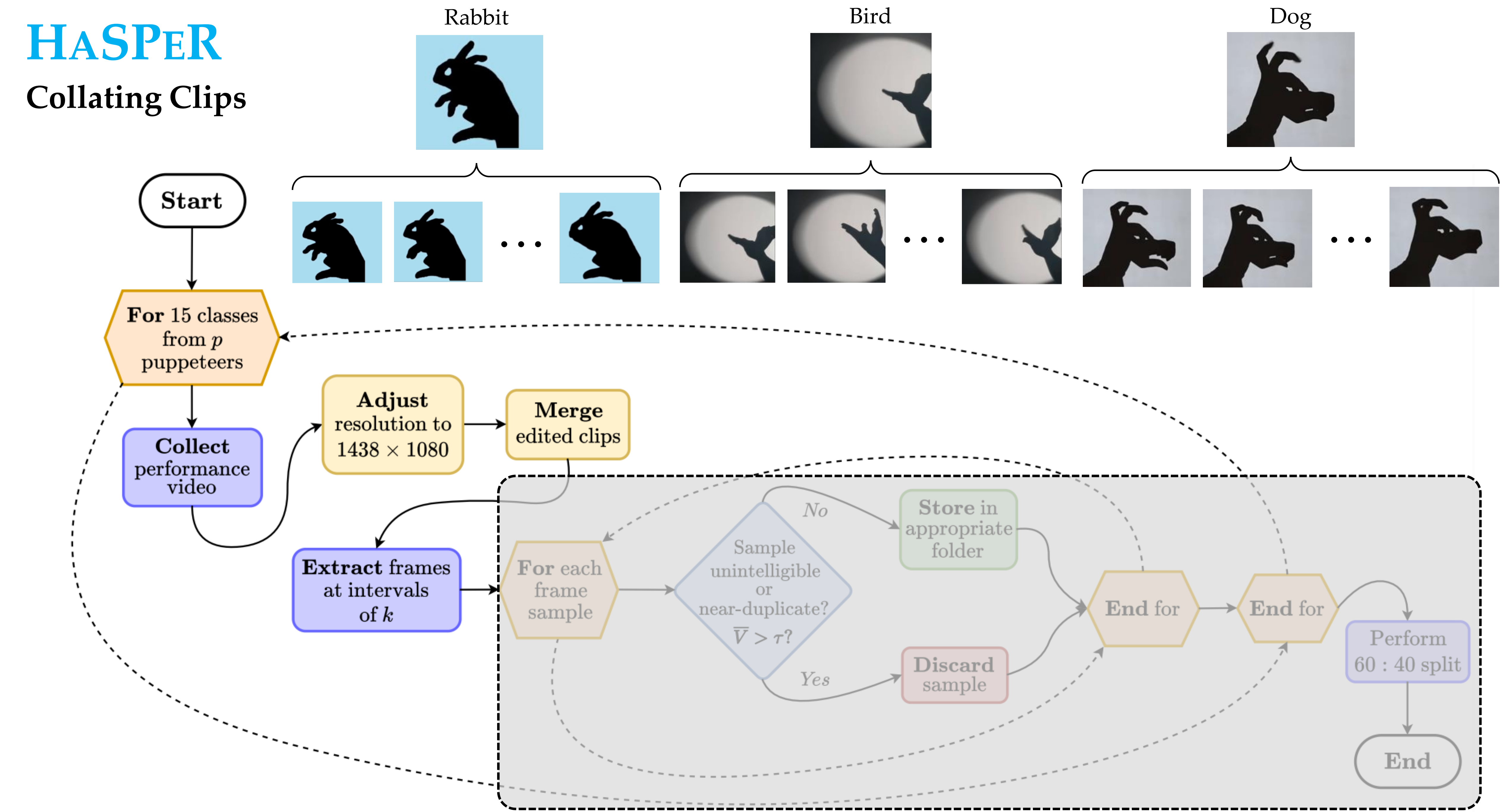
- **Benchmarking** — Established benchmarks for the dataset.
  - ✓ With **31** SoTA pre-trained Pytorch feature extractor architectures as baselines.
  - ✓ Found superiority of **skip-connected convolutional models** over **attention-based transformers models** in silhouette classification.
  - ✓ Experimented with feature fusion techniques
    - ➢ Topological features
    - ➢ Silhouette polygonization
- **Prototype Application** — Developed a lightweight Android app using Flutter for real-time classification of hand shadow puppets from camera feeds

HaSPeR

Dataset Construction Flowchart

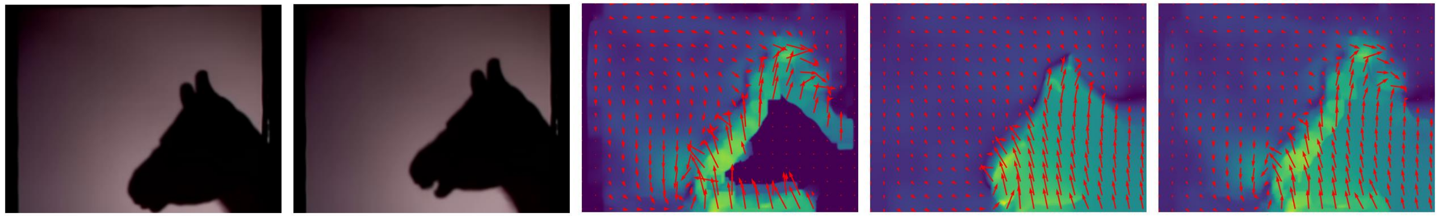


HaSPeR
Collating Clips



HaSPeR

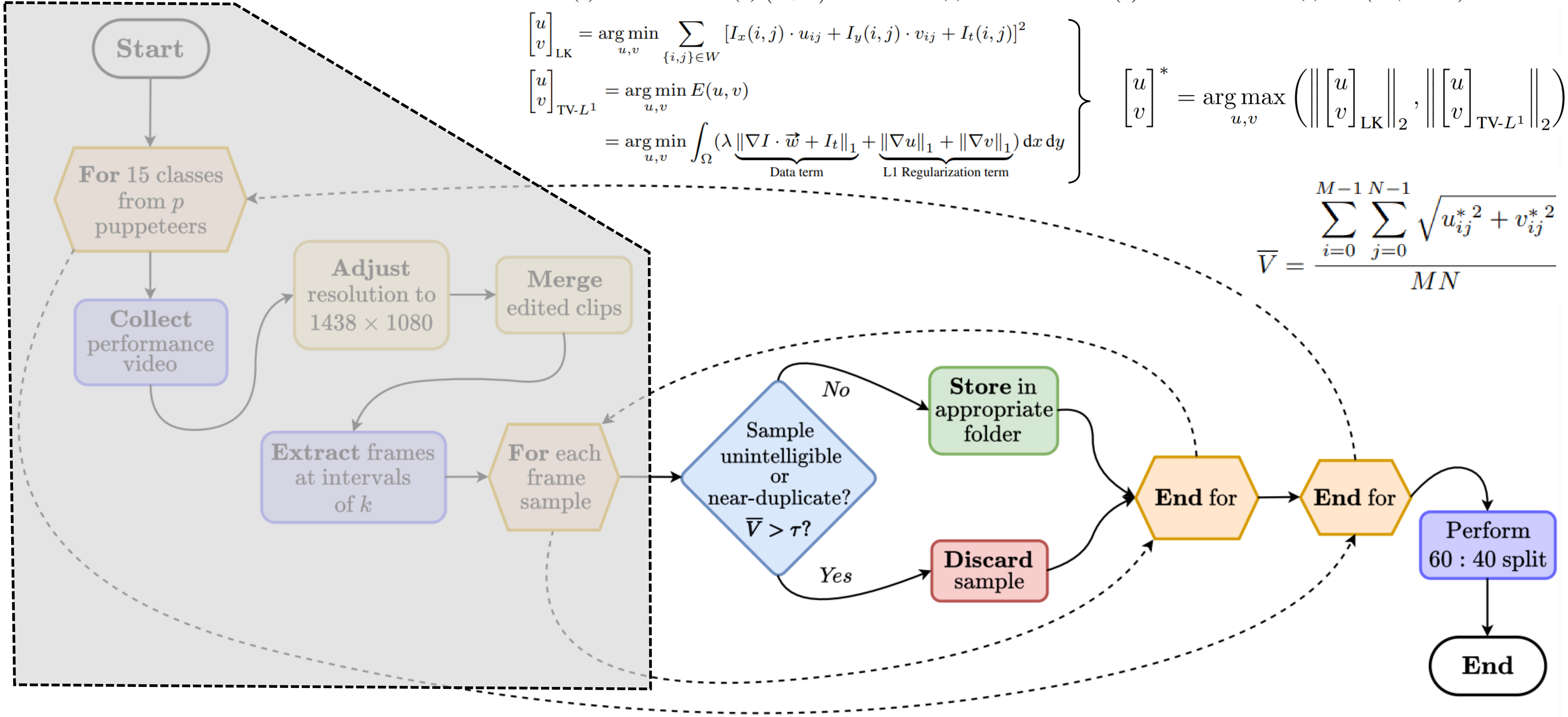
Optical Flow Estimation



(a) tth frame (b) $(t+k)$th frame (c) LK method (d) TV-L^1 method (e) $\max(\text{LK}, \text{TV-}L^1)$

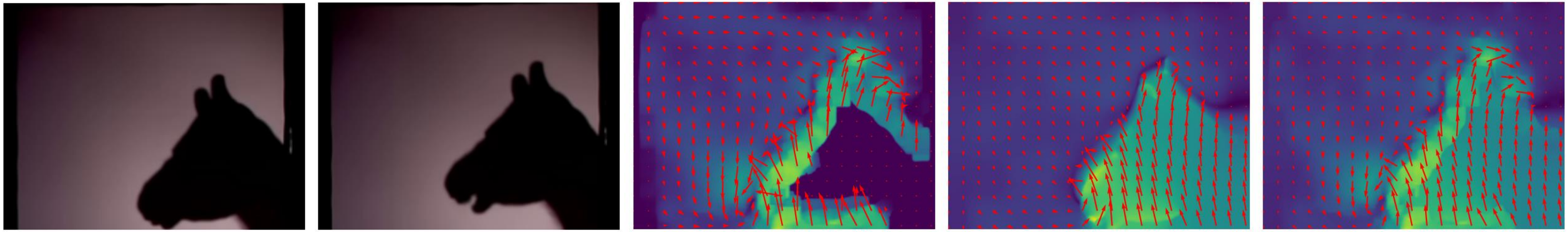
$$\left.\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} &= \arg\min_{u,v} \sum_{\{i,j\}\in W} [I_x(i,j)\cdot u_{ij} + I_y(i,j)\cdot v_{ij} + I_t(i,j)]^2 \\ \begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} &= \arg\min_{u,v} E(u,v) \\ &= \arg\min_{u,v} \int_\Omega (\lambda \underbrace{\|\nabla I \cdot \vec{w} + I_t\|_1}_{\text{Data term}} + \underbrace{\|\nabla u\|_1 + \|\nabla v\|_1}_{\text{L1 Regularization term}})\, dx\, dy \end{aligned}\right\} \quad \begin{bmatrix} u \\ v \end{bmatrix}^* = \arg\max_{u,v} \left(\left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} \right\|_2, \left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} \right\|_2 \right)$$

$$\overline{V} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sqrt{{u_{ij}^*}^2 + {v_{ij}^*}^2}}{MN}$$



HaSPeR

Optical Flow Estimation



(a) tth frame (b) $(t+k)$th frame (c) LK method [8] (d) TV-L^1 method [9] (e) $\max(\text{LK}, \text{TV-}L^1)$

$$\left.\begin{aligned}
\begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} &= \arg\min_{u,v} \sum_{\{i,j\}\in W} \left[I_x(i,j)\cdot u_{ij} + I_y(i,j)\cdot v_{ij} + I_t(i,j) \right]^2 \\
\begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} &= \arg\min_{u,v} E(u,v) \\
&= \arg\min_{u,v} \int_\Omega \Big(\lambda \underbrace{\|\nabla I \cdot \vec{w} + I_t\|_1}_{\text{Data term}} + \underbrace{\|\nabla u\|_1 + \|\nabla v\|_1}_{\text{L1 Regularization term}}\Big)\, dx\, dy
\end{aligned}\right\}
\begin{bmatrix} u \\ v \end{bmatrix}^* = \arg\max_{u,v} \left(\left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} \right\|_2, \left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} \right\|_2 \right)$$

$$\overline{V} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sqrt{{u_{ij}^*}^2 + {v_{ij}^*}^2}}{MN}$$

HaSPeR

Dataset Statistics

Silhouette Class	Clips		Sample Distribution		
	Pro.	Nov.	Training	Validation	Total
Bird	6	6	600	400	1000
Chicken	2	6	600	400	1000
Cow	2	6	600	400	1000
Crab	4	6	600	400	1000
Deer	6	6	600	400	1000
Dog	7	6	600	400	1000
Elephant	5	6	600	400	1000
Horse	8	6	600	400	1000
Llama	2	6	600	400	1000
Moose	3	6	600	400	1000
Panther	2	6	600	400	1000
Rabbit	4	6	600	400	1000
Snail	4	6	600	400	1000
Snake	3	6	600	400	1000
Swan	10	6	600	400	1000
Total	68	90	9000	6000	15000
	158				

Fig: Statistical summary of HaSPeR.

Fig: Samples from each class of the dataset.

- For each class, **professional** $\approx 47.827 \pm 1.414\%$ and **amateur** $\approx 52.172 \pm 1.414\%$
- Proportions of professionally sourced samples belonging to the 'Llama' and 'Snake' classes (14% and 27.3% respectively) are slightly low due to a **scarcity** of performance clips.

HaSPeR

Sample Diversity

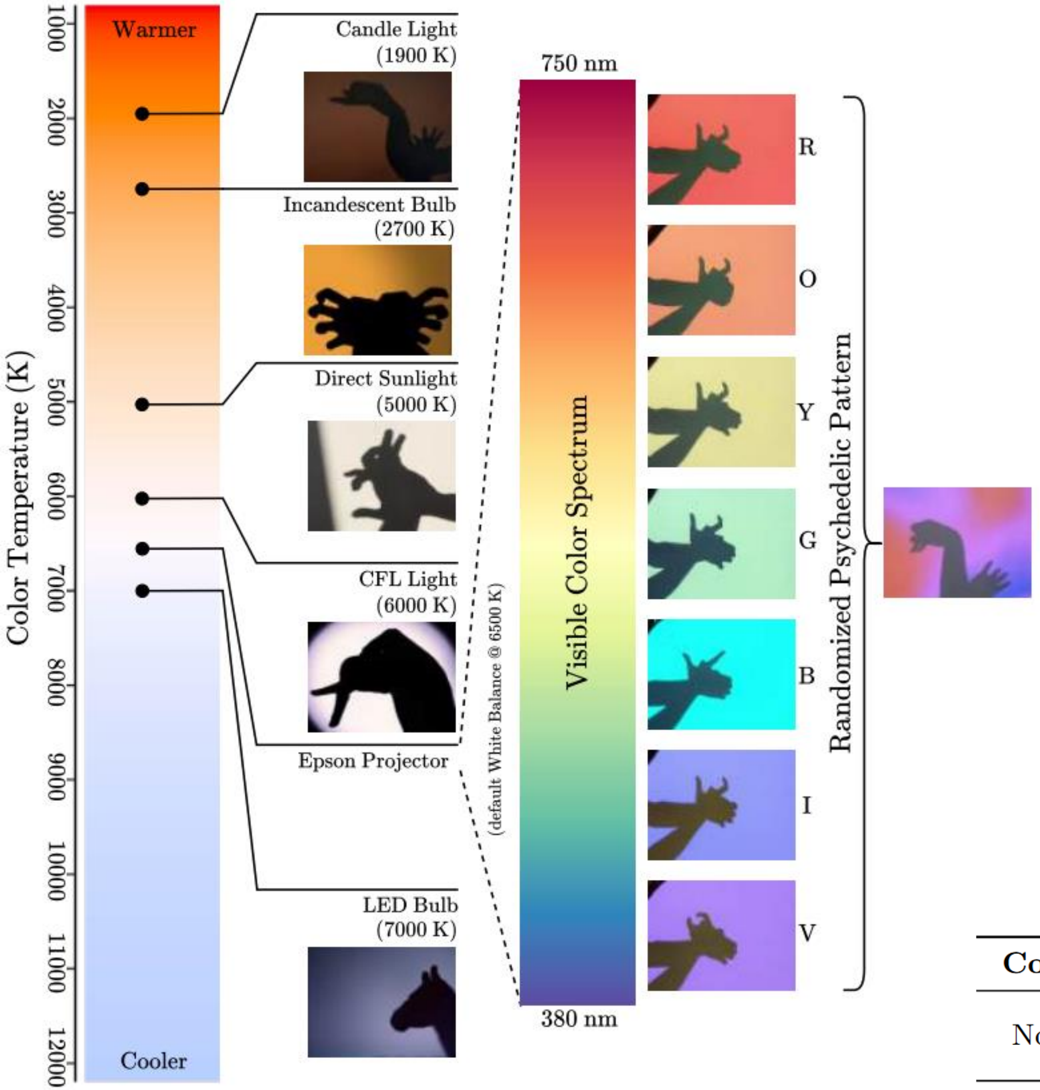


Fig: Light sources for background diversity.

(a) Sharp, high opacity (b) Diffuse, low opacity

Fig: Samples with different silhouette properties.

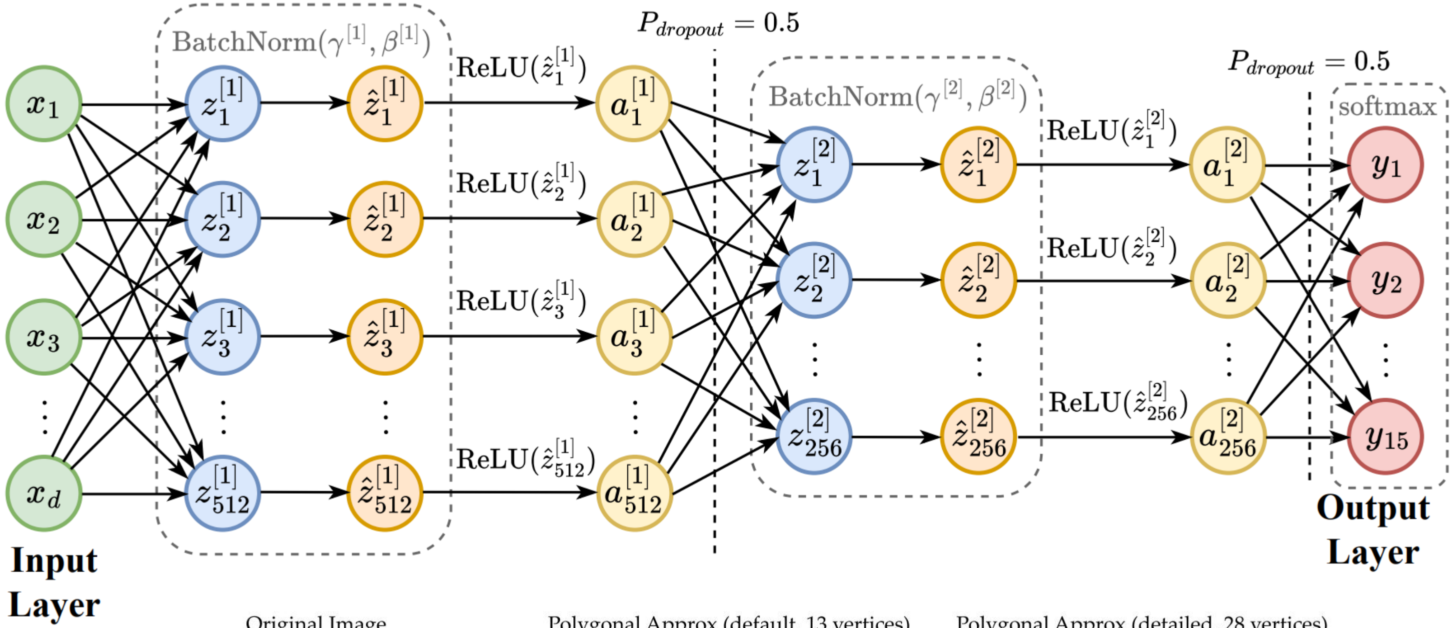
Fig: 'Deer' samples with different artistic representations.

Cohort	Subgroup	n	Gender (M:F)	Age Range	Hand Length (cm)	Hand Width (cm)
Novice	Adults Minors	6	3:3	9 to 25	18.75 ± 1.55 14.23 ± 1.16	8.66 ± 0.77 6.73 ± 0.82
Professional	–	14	12:2	N/A	N/A	N/A

Benchmarking

Models and Modifications

- **Feature Extractor Models — 31** SoTA baselines pre-trained on ImageNet
  - ✓ With a vanilla fully-connected layer
  - ✓ With a simple adapter block
  - ✓ With feature fusions (*concat*)
    - ➢ **Silhouette Polygonization [10]**
      - ▪ Vertex coordinates
    - ➢ **Topological Features [11]**
      - ▪ Betti curves
      - ▪ Morphological features
      - ▪ Local extrema coordinates
      - ▪ Euler characteristic
      - ▪ Gradient magnitude
      - ▪ Contours



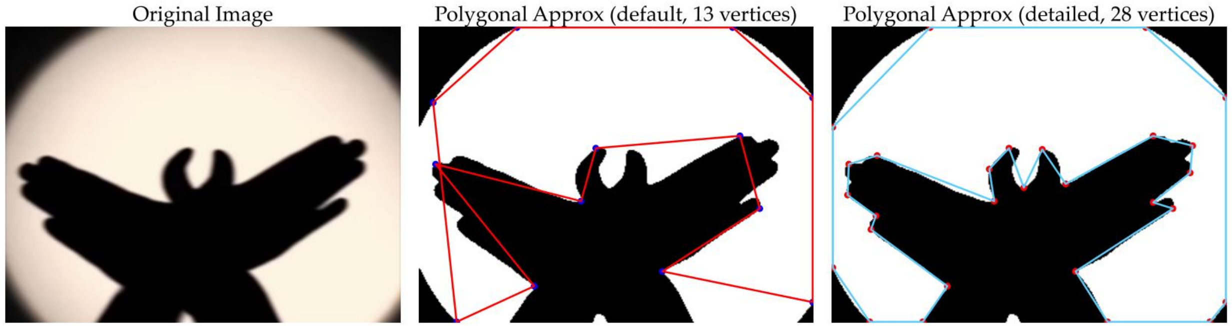


Fig: Polygonal approximations for a hand shadow puppet silhouette.

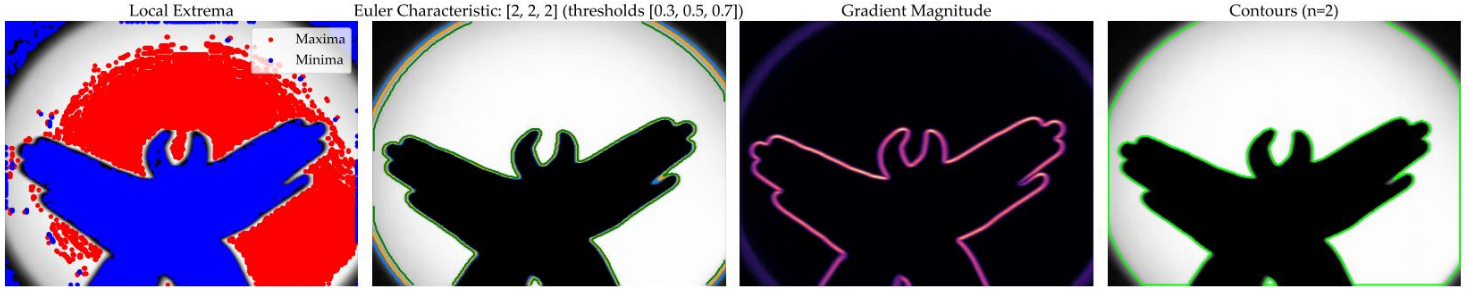


Fig: Topological features for a hand shadow puppet silhouette.

Results

Tentative Benchmarking Results

Metrics:
- Top-k accuracy
- Precision
- Recall
- F1-score

Hyperparameters:
- $\alpha = 0.001$
- $\gamma_{momentum} = 0.9$
- $\gamma_{decay} = 0.1$ per 5
- Epochs = 50

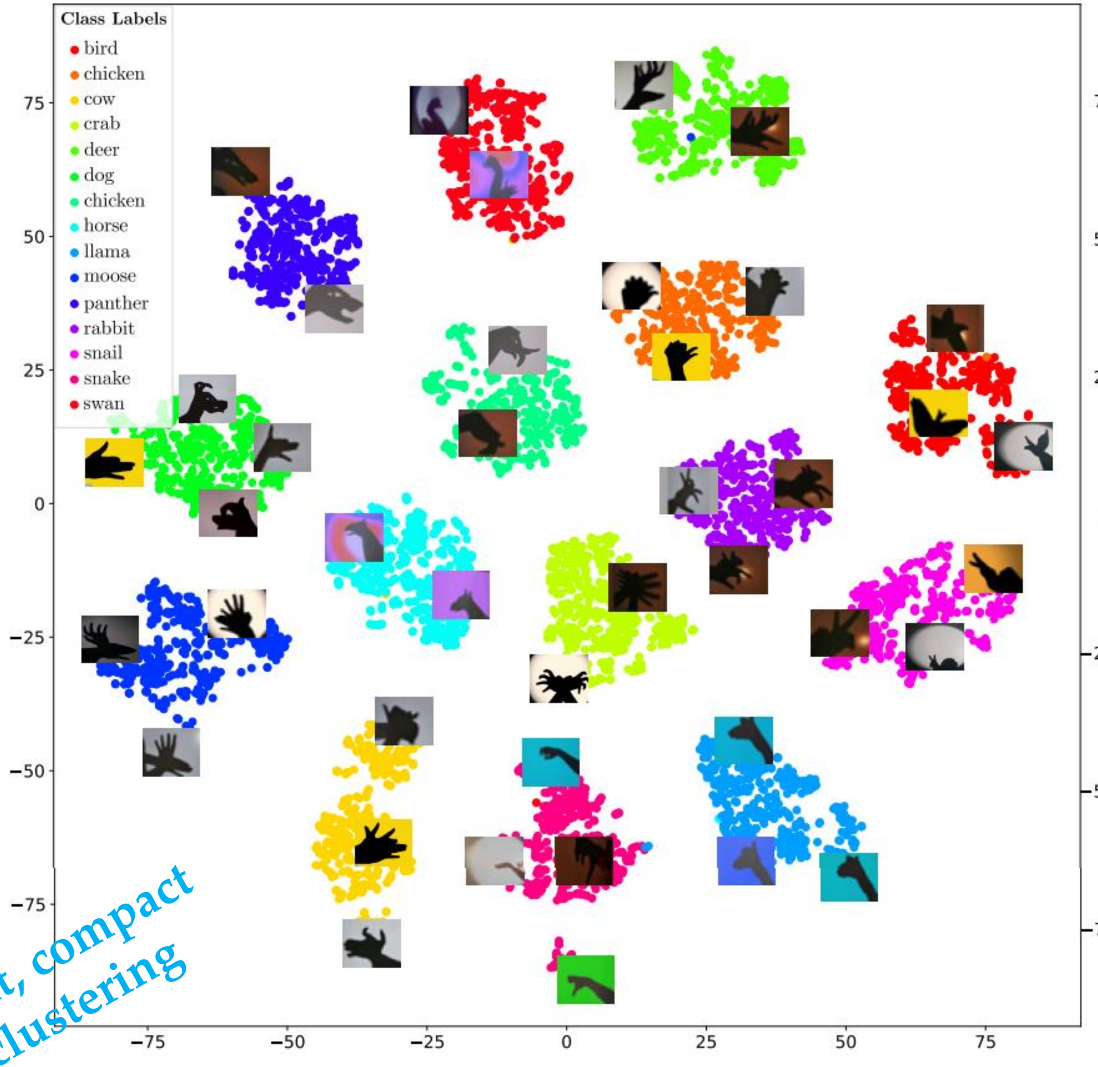
Models	Params.	Performance Metrics											
		Vanilla						w/ Classifier Block					
		Top-k Accuracy (%)			Precision	Recall	F1-score	Top-k Accuracy (%)			Precision	Recall	F1-score
		Top-1	Top-2	Top-3				Top-1	Top-2	Top-3			
ShuffleNetV2X10 [40]	2.3M	61.73	78.41	86.10	0.6559	0.6173	0.5970	88.73	93.98	96.10	0.8995	0.8873	0.8853
ViTB16 [41]	86.6M	69.71	77.60	83.28	0.7276	0.6972	0.6969	68.88	76.65	81.36	0.7192	0.6868	0.6851
ViTL32 [41]	306.5M	85.10	91.56	94.48	0.8720	0.8510	0.8509	84.71	91.80	94.08	0.8632	0.8472	0.8465
AlexNet [11]	61.1M	87.01	93.61	95.46	0.8840	0.8702	0.8708	88.18	92.58	94.80	0.8887	0.8818	0.8809
SqueezeNet1_1 [42]	1.2M	87.56	92.45	94.15	0.8880	0.8757	0.8744	86.21	92.48	94.65	0.8754	0.8622	0.8637
MobileNetV3Small [43]	2.5M	89.48	94.31	95.76	0.9038	0.8948	0.8942	89.85	94.35	96.48	0.9082	0.8985	0.8976
SwinB [44]	87.8M	90.50	95.38	97.40	0.9128	0.9050	0.9042	90.20	95.40	97.08	0.9097	0.902	0.9006
GoogLeNet [45]	6.6M	90.73	94.65	95.70	0.9105	0.9073	0.9059	92.18	95.65	96.58	0.9283	0.9218	0.9206
ResNet18 [46]	11.7M	90.91	95.28	96.60	0.9176	0.9092	0.9069	91.25	95.43	97.05	0.9229	0.9125	0.9119
MobileNetV3Large [43]	5.5M	91.20	94.48	95.98	0.9185	0.9120	0.9110	90.40	94.53	95.26	0.9147	0.9040	0.9024
ConvNeXt [47]	88.6M	91.46	96.33	98.05	0.9220	0.9147	0.9140	92.55	96.36	97.96	0.9306	0.9255	0.9246
SwinV2B [48]	87.9M	91.58	96.25	97.61	0.9210	0.9158	0.9151	91.48	96.00	97.55	0.9209	0.9148	0.9144
VGG16 [12]	138.4M	91.61	95.08	96.65	0.9248	0.9162	0.9168	91.00	95.21	96.45	0.9235	0.9100	0.9119
MNasNet13 [49]	6.3M	91.66	95.65	97.01	0.9240	0.9167	0.9149	91.45	95.86	97.26	0.9231	0.9145	0.9133
ConvNeXtLarge [47]	197.8M	91.88	95.90	97.70	0.9254	0.9188	0.9181	88.00	94.70	96.56	0.8942	0.8800	0.8782
EfficientNetB0 [50]	5.3M	91.93	95.26	96.71	0.9257	0.9193	0.9178	90.40	93.75	95.10	0.9131	0.9040	0.9022
MaxViT [51]	30.9M	92.01	96.50	97.81	0.9268	0.9202	0.9214	92.08	95.98	97.36	0.9320	0.9208	0.9237
EfficientNetV2S [52]	21.5M	92.31	95.75	96.76	0.9375	0.9232	0.9245	**94.45**	**97.35**	**98.30**	0.9498	**0.9445**	0.9438
VGG19 [12]	143.7M	92.36	95.13	96.10	0.9354	0.9237	0.9242	91.80	95.06	96.15	0.9296	0.9180	0.9187
MobileNetV2 [53]	3.5M	92.38	94.98	96.05	0.9303	0.9238	0.9233	92.31	95.38	96.91	0.9311	0.9232	0.9225
WideResNet50_2 [54]	68.9M	92.46	96.28	97.28	0.9331	0.9247	0.9235	93.35	95.73	97.15	0.9421	0.9335	0.9330
ResNet50 [46]	25.6M	92.58	95.56	96.75	0.9332	0.9258	0.9252	93.08	96.48	97.20	0.9363	0.9308	0.9299
RegNetX32GF [55]	107.8M	92.86	95.71	96.93	0.9348	0.9287	0.9269	92.91	95.71	96.95	0.9366	0.9292	0.9282
DenseNet121 [56]	8.0M	92.93	95.75	96.88	0.9367	0.9293	0.9282	92.95	95.51	96.56	0.9360	0.9295	0.9285
ResNeXt101_32X8D [57]	88.8M	93.00	96.41	97.23	0.9364	0.9310	0.9303	94.20	96.61	97.58	**0.9520**	0.9420	0.9423
WideResNet101_2 [54]	126.9M	93.36	95.81	96.90	0.9423	0.9337	0.9332	92.73	96.35	97.63	0.9337	0.9273	0.9267
InceptionV3 [58]	27.2M	93.50	96.48	97.35	0.9401	0.9350	0.9338	93.71	96.36	97.06	0.9446	0.9372	0.9371
DenseNet201 [56]	20.0M	93.56	95.78	96.73	0.9450	0.9357	0.9353	94.43	97.00	97.61	0.9492	0.9443	**0.9442**
ResNet101 [46]	44.5M	93.81	96.23	97.71	0.9432	0.9382	0.9406	93.23	96.93	98.13	0.9386	0.9323	0.9321
ResNet152 [46]	60.2M	94.06	97.06	98.05	0.9447	0.9407	0.9394	93.05	96.73	97.48	0.9374	0.9305	0.9297
ResNet34 [46]	21.8M	**94.97**	**97.23**	**98.23**	**0.9516**	**0.9497**	**0.9491**	91.98	95.95	97.20	0.9266	0.9198	0.9189
ResNet34 w/ Silhouette Polygonization	21.8M	92.72	96.41	97.51	0.9328	0.9272	0.9257	92.95 (+1.05%)	95.75	96.61	0.9352 (+0.93%)	0.9295 (+1.05%)	0.9283 (+1.02%)
ResNet34 w/ Topological Features	21.8M	93.72	96.43	97.78	0.9432	0.9372	0.9359	94.05 (+2.25%)	96.45 (+0.52%)	97.53 (+0.34%)	0.9476 (+2.27%)	0.9405 (+2.25%)	0.9401 (+2.31%)

Models	Params.	Performance Metrics											
		Vanilla						w/ Classifier Block					
		Top-k Accuracy (%)			Precision	Recall	F1-score	Top-k Accuracy (%)			Precision	Recall	F1-score
		Top-1	Top-2	Top-3				Top-1	Top-2	Top-3			
ShuffleNetV2X10 [40]	2.3M	61.73	78.41	86.10	0.6559	0.6173	0.5970	88.73	93.98	96.10	0.8995	0.8873	0.8853
ViTB16 [41]	86.6M	69.71	77.60	83.28	0.7276	0.6972	0.6969	68.88	76.65	81.36	0.7192	0.6868	0.6851
ViTL32 [41]	306.5M	85.10	91.56	94.48	0.8720	0.8510	0.8509	84.71	91.80	94.08	0.8632	0.8472	0.8465
AlexNet [11]	61.1M	87.01	93.61	95.46	0.8840	0.8702	0.8708	88.18	92.58	94.80	0.8887	0.8818	0.8809
SqueezeNet1_1 [42]	1.2M	87.56	92.45	94.15	0.8880	0.8757	0.8744	86.21	92.48	94.65	0.8754	0.8622	0.8637
MobileNetV3Small [43]	2.5M	89.48	94.31	95.76	0.9038	0.8948	0.8942	89.85	94.35	96.48	0.9082	0.8985	0.8976
SwinB [44]	87.8M	90.50	95.38	97.40	0.9128	0.9050	0.9042	90.20	95.40	97.08	0.9097	0.902	0.9006
GoogLeNet [45]	6.6M	90.73	94.65	95.70	0.9105	0.9073	0.9059	92.18	95.65	96.58	0.9283	0.9218	0.9206
ResNet18 [46]	11.7M	90.91	95.28	96.60	0.9176	0.9092	0.9069	91.25	95.43	97.05	0.9229	0.9125	0.9119
MobileNetV3Large [43]	5.5M	91.20	94.48	95.98	0.9185	0.9120	0.9110	90.40	94.53	95.26	0.9147	0.9040	0.9024
ConvNeXt [47]	88.6M	91.46	96.33	98.05	0.9220	0.9147	0.9140	92.55	96.36	97.96	0.9306	0.9255	0.9246
SwinV2B [48]	87.9M	91.58	96.25	97.61	0.9210	0.9158	0.9151	91.48	96.00	97.55	0.9209	0.9148	0.9144
VGG16 [12]	138.4M	91.61	95.08	96.65	0.9248	0.9162	0.9168	91.00	95.21	96.45	0.9235	0.9100	0.9119
MNasNet13 [49]	6.3M	91.66	95.65	97.01	0.9240	0.9167	0.9149	91.45	95.86	97.26	0.9231	0.9145	0.9133
ConvNeXtLarge [47]	197.8M	91.88	95.90	97.70	0.9254	0.9188	0.9181	88.00	94.70	96.56	0.8942	0.8800	0.8782
EfficientNetB0 [50]	5.3M	91.93	95.26	96.71	0.9257	0.9193	0.9178	90.40	93.75	95.10	0.9131	0.9040	0.9022
MaxViT [51]	30.9M	92.01	96.50	97.81	0.9268	0.9202	0.9214	92.08	95.98	97.36	0.9320	0.9208	0.9237
EfficientNetV2S [52]	21.5M	92.31	95.75	96.76	0.9375	0.9232	0.9245	**94.45**	**97.35**	**98.30**	0.9498	**0.9445**	0.9438
VGG19 [12]	143.7M	92.36	95.13	96.10	0.9354	0.9237	0.9242	91.80	95.06	96.15	0.9296	0.9180	0.9187
MobileNetV2 [53]	3.5M	92.38	94.98	96.05	0.9303	0.9238	0.9233	92.31	95.38	96.91	0.9311	0.9232	0.9225
WideResNet50_2 [54]	68.9M	92.46	96.28	97.28	0.9331	0.9247	0.9235	93.35	95.73	97.15	0.9421	0.9335	0.9330
ResNet50 [46]	25.6M	92.58	95.56	96.75	0.9332	0.9258	0.9252	93.08	96.48	97.20	0.9363	0.9308	0.9299
RegNetX32GF [55]	107.8M	92.86	95.71	96.93	0.9348	0.9287	0.9269	92.91	95.71	96.95	0.9366	0.9292	0.9282
DenseNet121 [56]	8.0M	92.93	95.75	96.88	0.9367	0.9293	0.9282	92.95	95.51	96.56	0.9360	0.9295	0.9285
ResNeXt101_32X8D [57]	88.8M	93.00	96.41	97.23	0.9364	0.9310	0.9303	94.20	96.61	97.58	**0.9520**	0.9420	0.9423
WideResNet101_2 [54]	126.9M	93.36	95.81	96.90	0.9423	0.9337	0.9332	92.73	96.35	97.63	0.9337	0.9273	0.9267
InceptionV3 [58]	27.2M	93.50	96.48	97.35	0.9401	0.9350	0.9338	93.71	96.36	97.06	0.9446	0.9372	0.9371
DenseNet201 [56]	20.0M	93.56	95.78	96.73	0.9450	0.9357	0.9353	94.43	97.00	97.61	0.9492	0.9443	**0.9442**
ResNet101 [46]	44.5M	93.81	96.23	97.71	0.9432	0.9382	0.9406	93.23	96.93	98.13	0.9386	0.9323	0.9321
ResNet152 [46]	60.2M	94.06	97.06	98.05	0.9447	0.9407	0.9394	93.05	96.73	97.48	0.9374	0.9305	0.9297
ResNet34 [46]	21.8M	**94.97**	**97.23**	**98.23**	**0.9516**	**0.9497**	**0.9491**	91.98	95.95	97.20	0.9266	0.9198	0.9189
ResNet34 w/ Silhouette Polygonization	21.8M	92.72	96.41	97.51	0.9328	0.9272	0.9257	92.95 (+1.05%)	95.75	96.61	0.9352 (+0.93%)	0.9295 (+1.05%)	0.9283 (+1.02%)
ResNet34 w/ Topological Features	21.8M	93.72	96.43	97.78	0.9432	0.9372	0.9359	94.05 (+2.25%)	96.45 (+0.52%)	97.53 (+0.34%)	0.9476 (+2.27%)	0.9405 (+2.25%)	0.9401 (+2.31%)

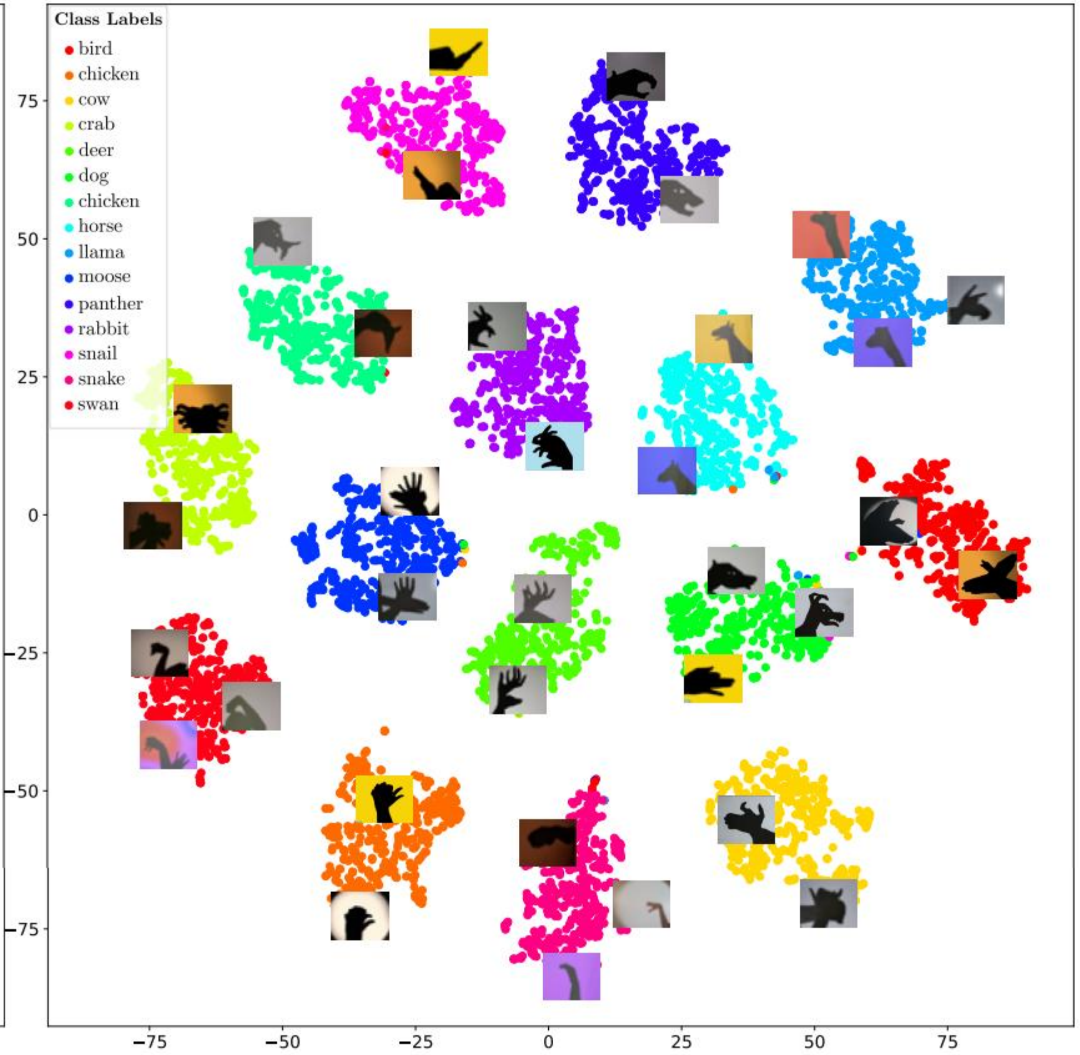
Results

Qualitative Analysis

Feature Space Visualization: *t*-SNE (*t*-Stochastic Neighbor Embedding)



(a) Vanilla ResNet34



(b) ResNet34 with Classifier Block

tight, compact clustering

Results

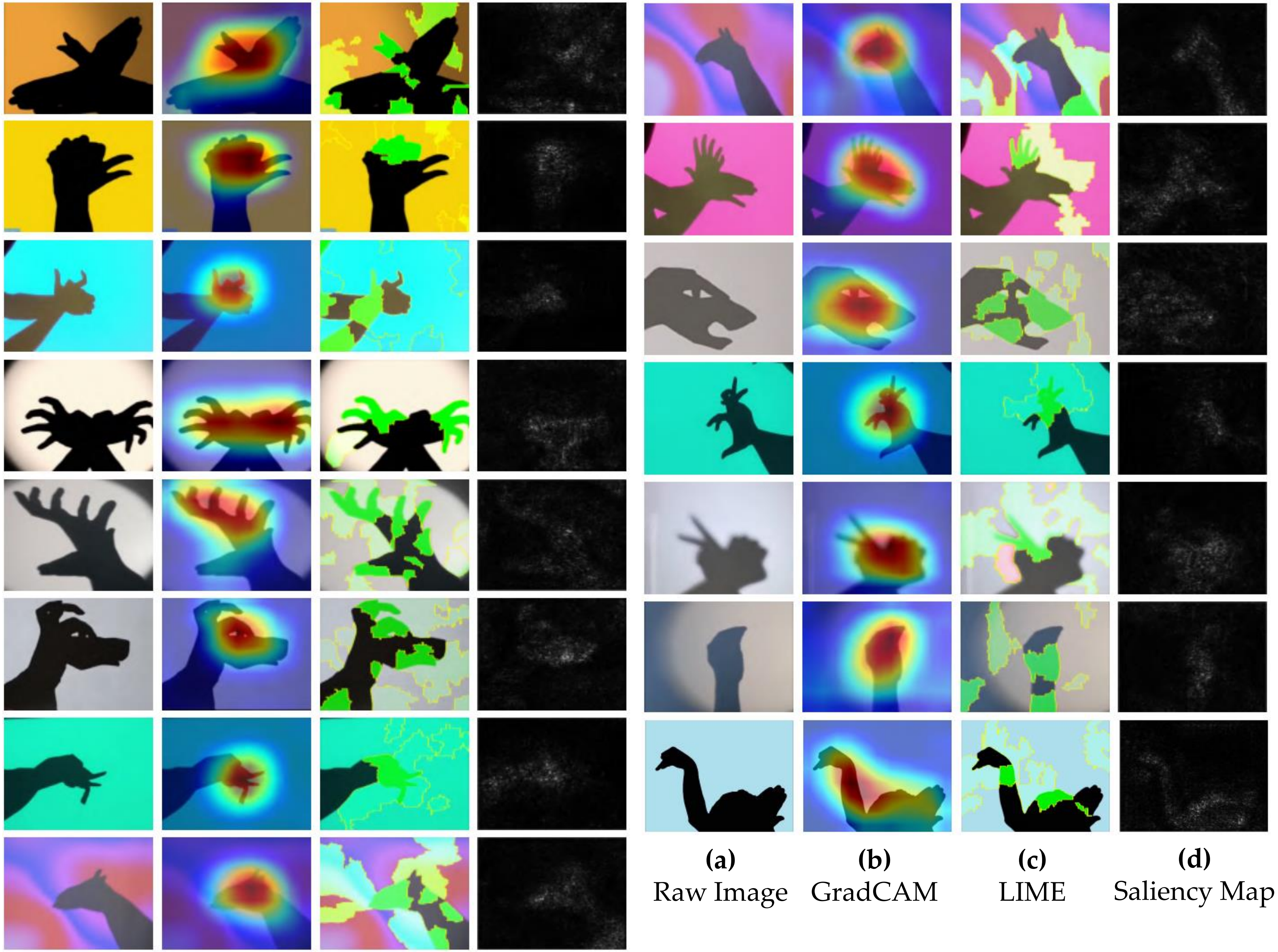
Qualitative Analysis

Explainable AI (xAI):

- GradCAM [12]
- LIME [13]
- Saliency Map [14]

Common-sense distinguishing features

- ✓ **Bird** wingspan, beak
- ✓ **Chicken** gallinaceous comb
- ✓ **Cow** horn, concave head
- ✓ **Crab** appendages
- ✓ **Deer** horns
- ✓ **Dog** slanted head, ears
- ✓ **Elephant** tusks, trunk
- ✓ **Moose** upright horns
- ✓ **Panther** eyes and ears
- ✓ **Rabbit** small hands and mouth
- ✓ **Snail** shell, antennae, etc.



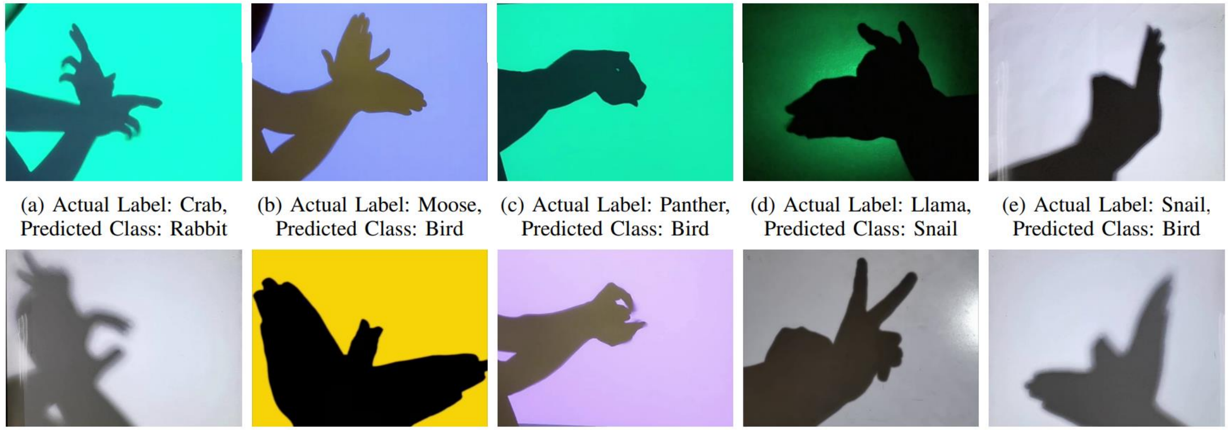
(a) Raw Image **(b)** GradCAM **(c)** LIME **(d)** Saliency Map

Results

Error Analysis

Probable reasons for misclassifications:

- High Inter-Class Similarity
- Significant Intra-Class Variation
- Ambiguity of shape present in mid-action frames
- Poor lighting and ineptitude of the amateur child puppeteers



(a) Actual Label: Crab, Predicted Class: Rabbit
(b) Actual Label: Moose, Predicted Class: Bird
(c) Actual Label: Panther, Predicted Class: Bird
(d) Actual Label: Llama, Predicted Class: Snail
(e) Actual Label: Snail, Predicted Class: Bird
(f) Similar sample from the 'Rabbit' class
(g) Similar sample from the 'Bird' class
(h) Similar sample from the 'Bird' class
(i) Similar sample from the 'Snail' class
(j) Similar sample from the 'Bird' class

Fig: Misclassified samples with visually similar samples of the predicted class.

True Labels \ Predicted Labels	Bird	Chicken	Cow	Crab	Deer	Dog	Elephant	Horse	Llama	Moose	Panther	Rabbit	Snail	Snake	Swan
Bird	386	0	6	0	0	0	0	0	0	8	0	0	0	0	0
Chicken	2	396	1	0	0	0	0	0	0	0	0	0	0	1	0
Cow	1	0	378	4	0	0	4	0	1	3	2	0	6	0	1
Crab	18	1	1	323	0	2	0	0	2	23	0	30	0	0	0
Deer	0	0	3	0	389	0	0	0	0	8	0	0	0	0	0
Dog	2	0	0	0	0	391	0	1	4	0	0	0	2	0	0
Elephant	3	0	0	0	0	0	395	0	0	0	0	0	2	0	0
Horse	0	0	1	0	0	4	0	395	0	0	0	0	0	0	0
Llama	0	0	0	0	0	0	0	0	388	0	0	0	12	0	0
Moose	6	1	0	4	2	0	0	0	0	387	0	0	0	0	0
Panther	17	0	0	0	0	5	11	22	6	0	322	1	0	15	1
Rabbit	1	1	0	4	0	0	1	0	0	0	0	390	2	0	1
Snail	20	3	10	0	0	3	0	0	0	1	1	0	362	0	0
Snake	0	0	0	0	0	0	0	0	0	0	0	0	0	399	1
Swan	0	0	0	1	0	2	0	0	0	0	0	0	0	0	397

Fig: Confusion Matrix of vanilla ResNet34.

Application: Digitization of Hand Shadow Puppetry

Mobile App Prototype

Objective: To create a real-time, interactive application that brings hand shadow puppetry to life using a lightweight trained model like MobileNetV2.

- Memory footprint: 29 MB
- Inference time: 880 μs

Snapdragon 8 Gen 2 of Samsung Galaxy S23

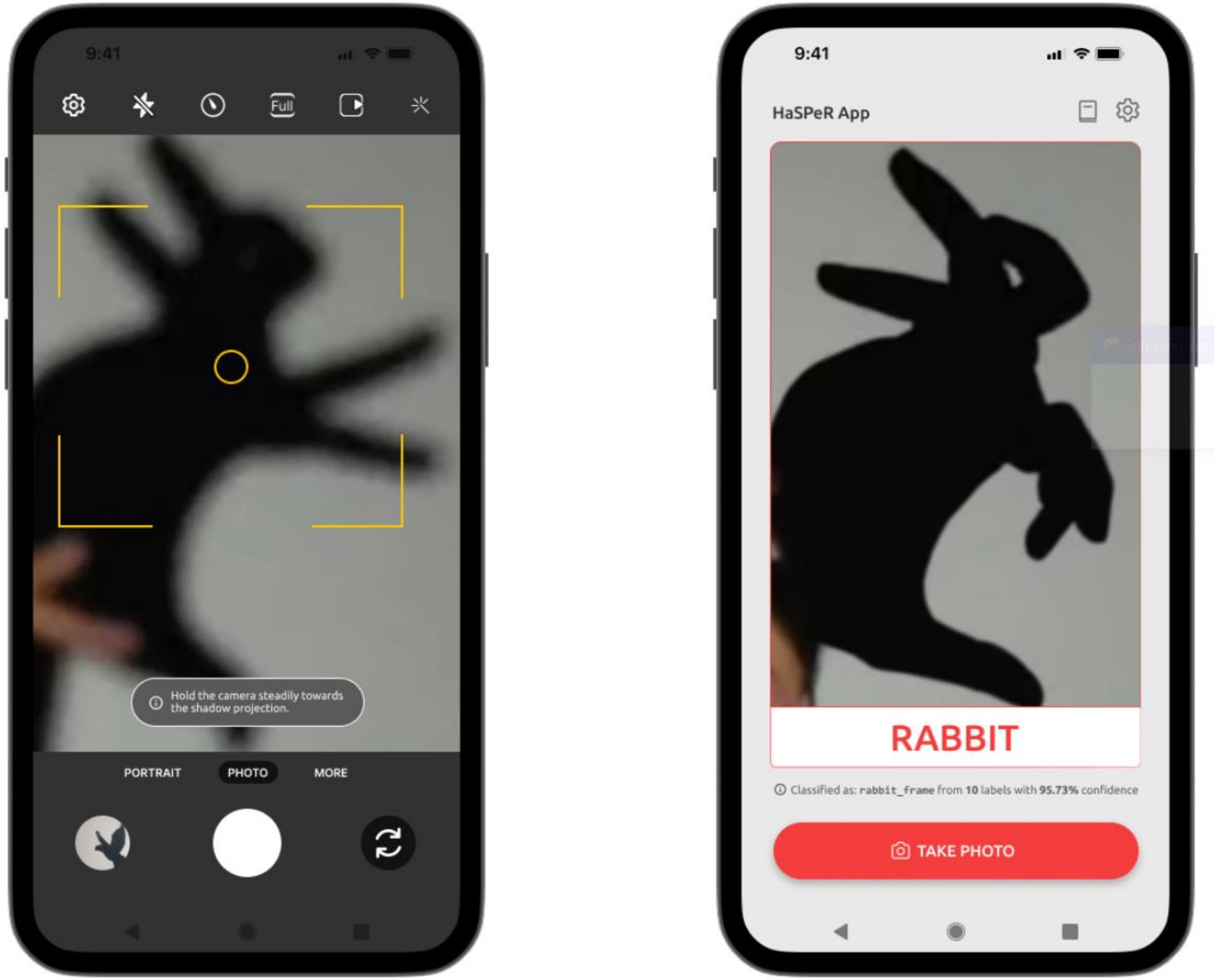
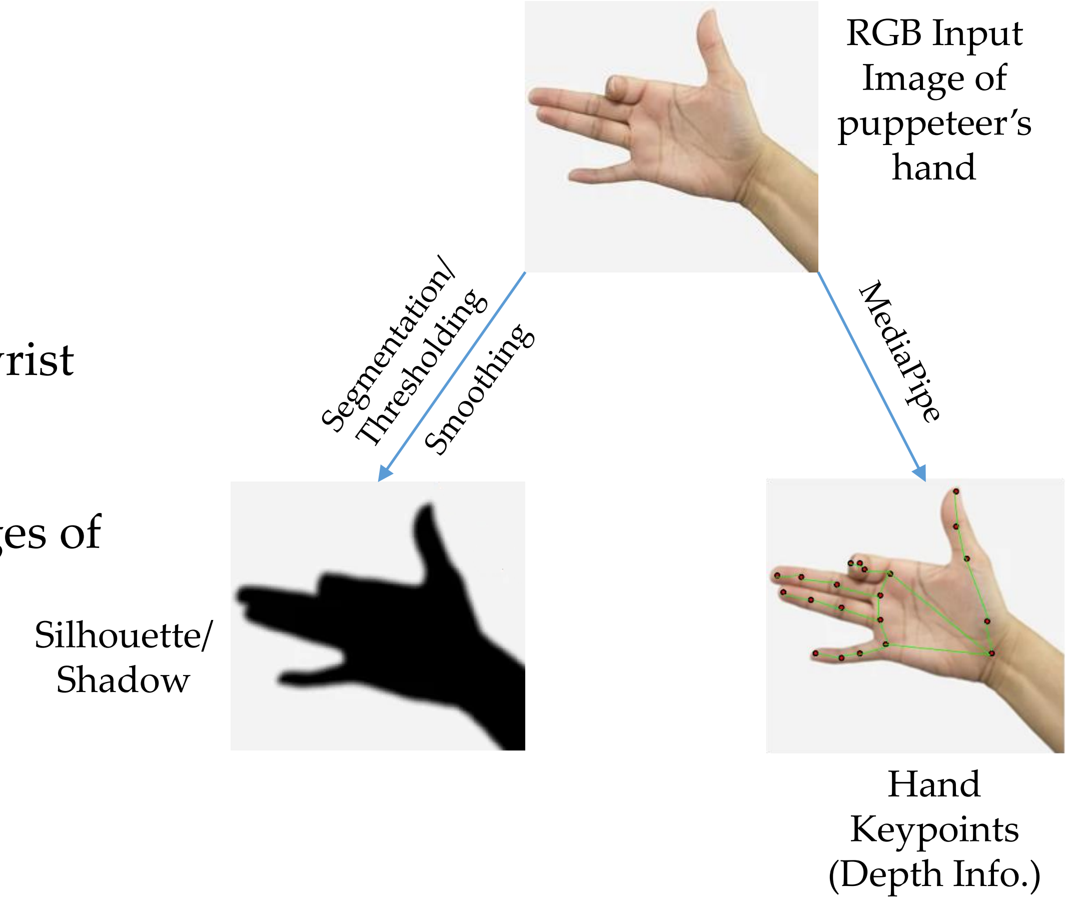


Fig: Android application for shadow puppet recognition.

Limitations

Scopes of Improvement

- Our work still has some ground to build upon:
  - ✓ Introducing samples with **more diversified** hand/palm/wrist structures.
  - ✓ Exploring **two different approaches** to classify RGB images of the hand
    - ➢ Feature Extraction after **RGB to grayscale silhouette conversion** (using pre-processing DIP techniques).
    - ➢ Utilizing **depth information** and coordinates of **hand landmarks** as features (using MediaPipe).
      - ✓ Yields **high accuracy in Sign Language Recognition** tasks, as per recently published research works.
  - ✓ Working on image/video generation.



Conclusion

Summary of our contributions

- We introduce **HASPER** (**Ha**nd **S**hadow **P**upp**e**t Image **R**epository), a novel, curated dataset of 15,000 images sourced from 68 professional and 90 amateur clips.
- We ensure diversity through variations in poses, orientations, background lighting, and silhouette motion via **optical flow estimation**.
- We evaluate **31 state-of-the-art pretrained image classification models** on HASPER to establish integrity baselines.
- We thoroughly assess **ResNet34**'s feature representations, feature fusions, interpretability, explainability, and classification errors.
- We develop a **lightweight Android app** using Flutter for real-time classification of hand shadow puppets from camera feeds, showcasing potential for digitized ombromanie learning tools.
- Our core finding is: **Skip-connected Convolutional models > Attention-based Transformers models**. (skip-connections help **preserve** low-level edge and contour information through **identity mappings**)

References

Works cited in this presentation

[1] A. Almoznino and Y. Pinas, "The art of hand shadows," pp. 1–64, 2002.

[2] F. Lu, F. Tian, Y. Jiang, X. Cao, W. Luo, G. Li, X. Zhang, G. Dai, and H. Wang, "Shadowstory: creative and collaborative digital storytelling inspired by cultural heritage," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. Vancouver BC, Canada: ACM, May 2011, p. 1919–1928.

[3] Z. Huang, V. K. Madaram, S. Albadrani, and T. V. Nguyen, "Shadow puppetry with robotic arms," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1251–1252, 2017.

[4] H. Zhang, Y. Song, Z. Chen, J. Cai, and K. Lu, "Chinese shadow puppetry with an interactive interface using the kinect sensor," in *Computer Vision–ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pp. 352–361, Springer, 2012.

[5] B. M. Carr and G. J. Brown, "Shadow puppetry using the kinect," 2014.

[6] M. Huang, S. Mehrotra, and F. Sparacino, "Shadow vision," 1999.

[7] https://www.kaggle.com/datasets/deepshah16/silhouettes-of-human-posture

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, vol. 2. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[9] C. Zach, T. Pock, and H. Bischof, A Duality Based Approach for Realtime TV-L 1 Optical Flow. *Springer Berlin Heidelberg*, p. 214–223. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74936-3 22

References

Works cited in this presentation

[10] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973. [Online]. Available: https://doi.org/10.3138/FM57-6770-U75U-7727

[11] H. Blum, "A Transformation for Extracting New Descriptors of Shape," in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. Cambridge, MA: The MIT Press, 1967, pp. 362–380.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, Oct. 2017, p. 618–626.

[13] M. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, J. DeNero, M. Finlayson, and S. Reddy, Eds. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 97–101.

[14] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: visualising image classification models and saliency maps," in *Proceedings of the ICLR*, 2014.

THANK YOU FOR LISTENING.

GET IN TOUCH:
EMAIL — {rifatraiyan, zibranzarif, sabbirahmed}@iut-dhaka.edu
TWITTER — @realRifatRaiyan

HASPER