HaSPeR: An Image Repository for Hand Shadow Puppet Recognition

ISLAMIC UNIVERSITY OF TECHNOLOGY

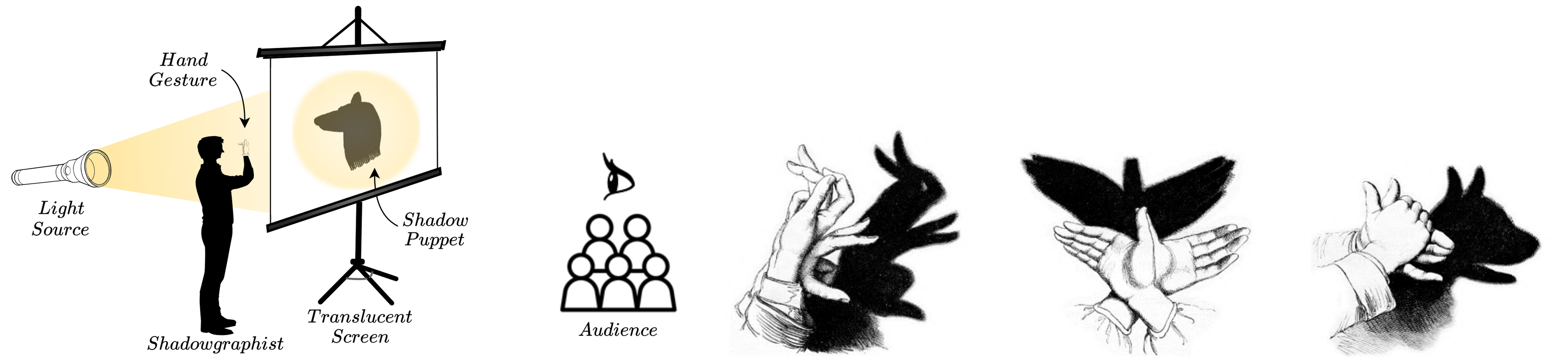
Syed Rifat Raiyan Zibran Zarif Amio Sabbir Ahmed

Department of Computer Science and Engineering, Islamic University of Technology, Board Bazar, Gazipur, Dhaka-1704, Bangladesh

ICCV HONOLULU HAWAII OCT 19-23, 2025

Introduction

Ombromanie, the ancient art of hand shadow puppetry, is a form of art that involves the mesmerizing interplay of light and shadow through the construction and manipulation of shadow figures or silhouettes on a surface, typically a screen or a wall, using one's hands, body, or props. The alias *"cinema in silhouette"* is sometimes used to refer to this proto-cinematic medium of entertainment.



(a) A generic hand shadow puppetry setup. (b) Rabbit (c) Bird (d) Dog

Figure 1. Ombromanie in a nutshell.

Hand Shadow Puppetry

Its working principle is very straightforward—the puppeteer adeptly positions their hands between a radiant light source and a translucent screen, consequently conjuring shadows and silhouettes that emulate different creatures, as shown in Figure 1.

Despite its rich history and cultural allure, ombromanie (hand shadow puppetry) suffers from a scarcity of specialized resources. Annotated datasets could enable detailed analysis of silhouette movements, shapes, and storytelling, fostering AI systems for recognition, classification, and generation of performances. This is especially pertinent given AI models' struggles with accurately rendering hands and fingers. Educational applications could revitalize this fading art, recognized by UNESCO in 2011 as an endangered intangible cultural heritage, underscoring the need for enhanced preservation and research efforts.

Contributions

In this paper [1], our contributions are:

- We introduce HaSPeR (**Ha**nd **S**hadow **P**uppet Image **R**epository), a novel, curated dataset of 15,000 images sourced from 68 professional and 90 amateur clips, with precise labeling and categorization for robust image classification training.
- We ensure diversity through variations in poses, orientations, background lighting, and silhouette motion via optical flow estimation.
- We conduct a detailed analysis of HaSPeR's statistical characteristics to validate its quality.
- We evaluate 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.

Dataset Construction

The series of steps involved in our data acquisition process is broadly divided into three tasks—(a) procuring the performance clips, (b) extraction of the frames, and (c) categorization of each sample frame with a proper label. Figure 2 portrays this workflow behind our dataset preparation. We incorporate manual oversight at each step of the dataset creation in order to reconcile any exigencies pertaining to the quality of HaSPeR.

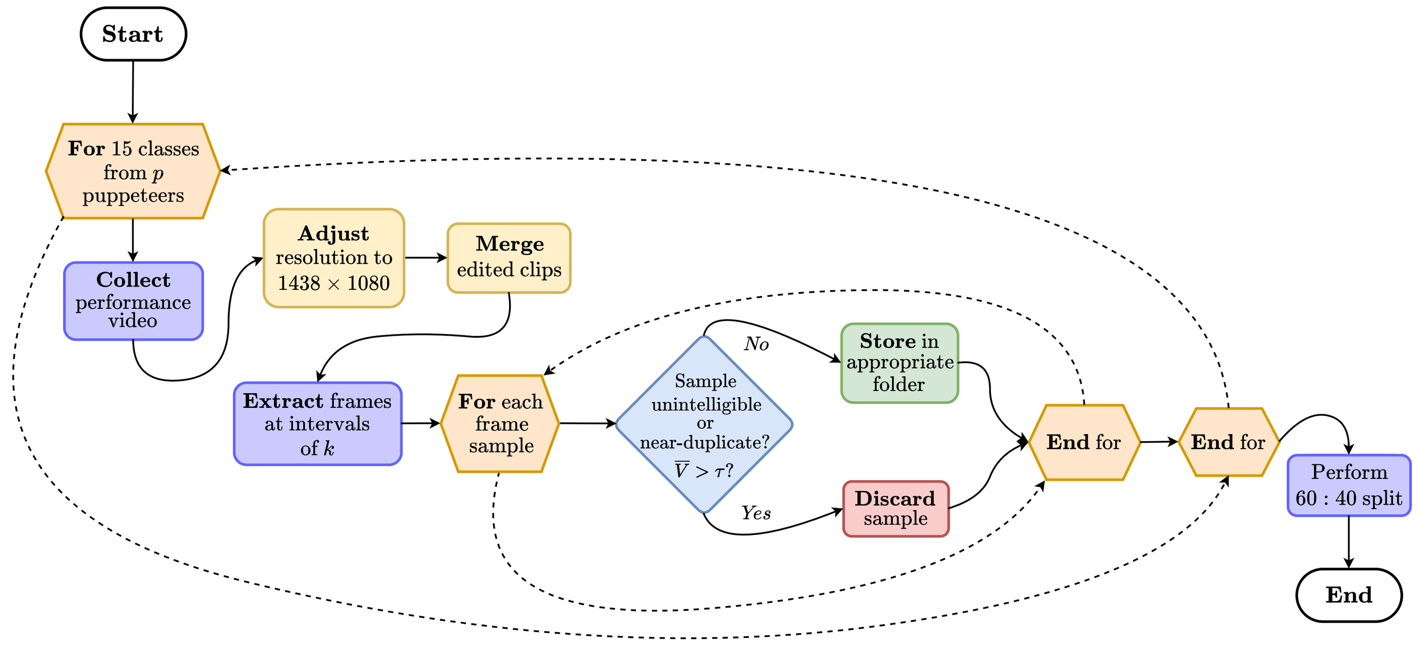


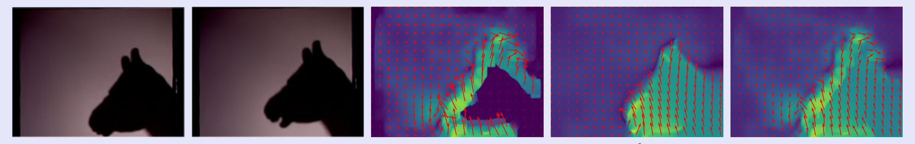
Figure 2. A flowchart depicting the dataset construction process.

(a) Bird (b) Chicken (c) Cow (d) Crab (e) Deer (f) Dog (g) Elephant (h) Horse

(i) Llama (j) Moose (k) Panther (l) Rabbit (m) Snail (n) Snake (o) Swan

Figure 3. Samples from each class of the dataset.

Optical Flow Estimation for Sample Extraction



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

Figure 4. Optical flow estimation of contiguous candidate frames from the 'Horse' class.

We retain the frame pairs with an average flow magnitude surpassing a certain requisite threshold τ, thereby ensuring the inclusion of dynamically distinct frames. We synergistically amalgamate two optical flow estimation methods: the Lucas–Kanade (LK) method and the Total Variation L1 Regularization (TV-L^1) method.

$$I_x \cdot u + I_y \cdot v + I_t = 0 \tag{1}$$

$$\begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} = \arg\min_{u,v} \sum_{\{i,j\}\in W} \left[I_x(i,j)\cdot u_{ij} + I_y(i,j)\cdot v_{ij} + I_t(i,j)\right]^2 \tag{2}$$

$$\begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} = \begin{bmatrix} \sum\limits_{\{i,j\}\in W} I_x^2 & \sum\limits_{\{i,j\}\in W} I_x I_y \\ \sum\limits_{\{i,j\}\in W} I_x I_y & \sum\limits_{\{i,j\}\in W} I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum\limits_{\{i,j\}\in W} I_x I_t \\ -\sum\limits_{\{i,j\}\in W} I_y I_t \end{bmatrix} \tag{3}$$

As evident from Figure 4c, the LK method can track the edge movement of the homogeneous silhouette patches, given the slight texture offered by the penumbral region of the shadow. However, as it is limited by local window constraints, it fails to capture the global motion of the shadow puppet. To ameliorate this issue, we resort to the TV-L^1 method, which is a variational method that minimizes the total variation of the flow field, subject to the L1 norm of the data fidelity term, which together form the energy function E in Equation 4.

$$\begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} = \arg\min_{u,v} E(u,v) \tag{4}$$

$$= \arg\min_{u,v} \int_\Omega \Big(\lambda \underbrace{\|\nabla I \cdot \vec{w} + I_t\|_1}_{\text{Data term}} + \underbrace{\|\nabla u\|_1 + \|\nabla v\|_1}_{\text{L1 Regularization term}}\Big)\,\mathrm{d}x\,\mathrm{d}y \tag{5}$$

We then take the element-wise maximum of the LK and TV-L^1 flow fields' L2 norms to obtain a more hoilistic optical flow field, as portrayed in Figure 4e.

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = \arg\max_{u,v} \left(\left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{LK}} \right\|_2, \left\| \begin{bmatrix} u \\ v \end{bmatrix}_{\text{TV-}L^1} \right\|_2 \right) \tag{6}$$

$$\overline{V} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sqrt{{u^*_{ij}}^2 + {v^*_{ij}}^2}}{MN} \tag{7}$$

If $\overline{V} > \tau$, we retain the corresponding frame pair as candidate samples, otherwise we continue the process with the $(t+k)$th and the $(t+2k)$th frames.

Dataset Description

The samples in HaSPeR vary in terms of the nature of the backgrounds, the anatomical structure of the puppeteers' hands, the photometric opacity and sharpness of the projected silhouettes, and a panoply of other aspects.

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				

Table 1. Statistical summary of HaSPeR.

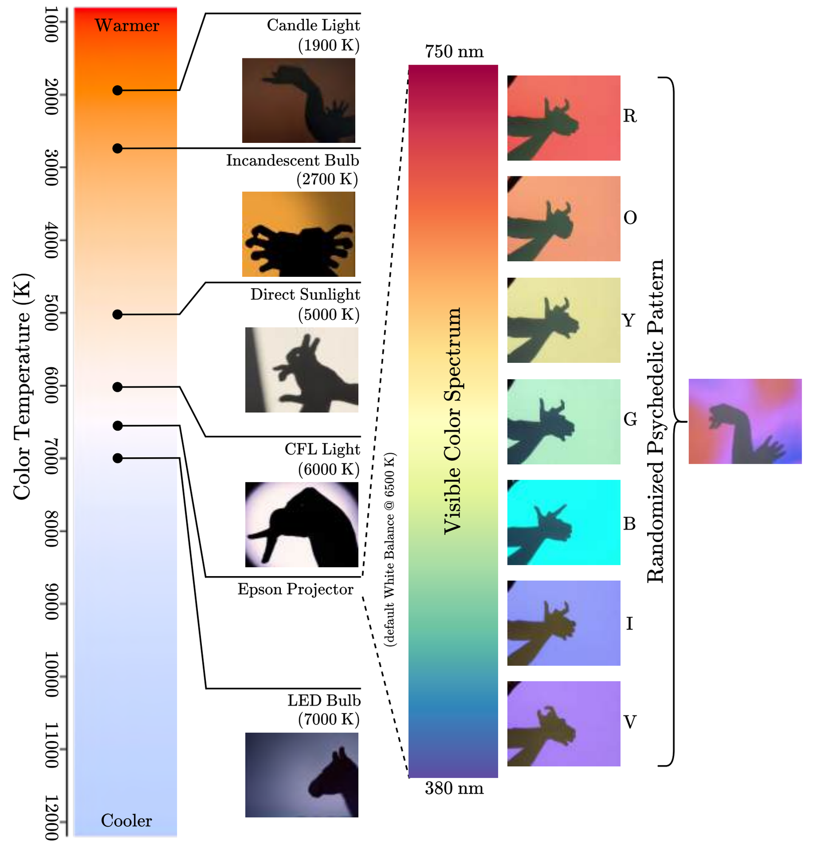


Figure 5. Light sources for background diversity in HaSPeR.

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

Figure 6. Samples with different silhouette properties.

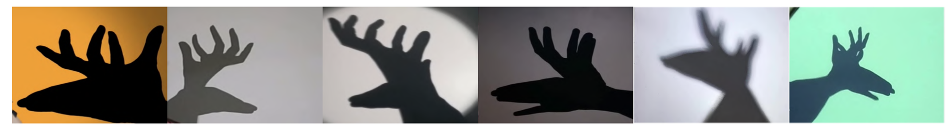


Figure 7. 'Deer' samples with different artistic representations.

Methodology for Benchmarking HaSPeR

A series of pretrained models are used as feature extractors to develop a benchmark for the dataset. The models are pretrained on the ImageNetdataset and fine-tuned on HaSPeR.

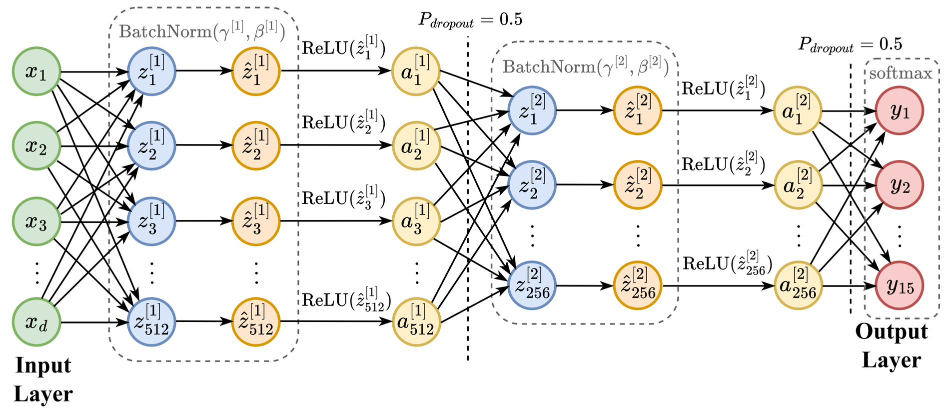


Figure 8 Classifier block attached to the tail-end of the pretrained models.

We augment ResNet34 with handcrafted polygonal features extracted from the silhouette contours of hand shadow puppets, using Douglas-Peucker polygonal approximation and geometric shape descriptors.

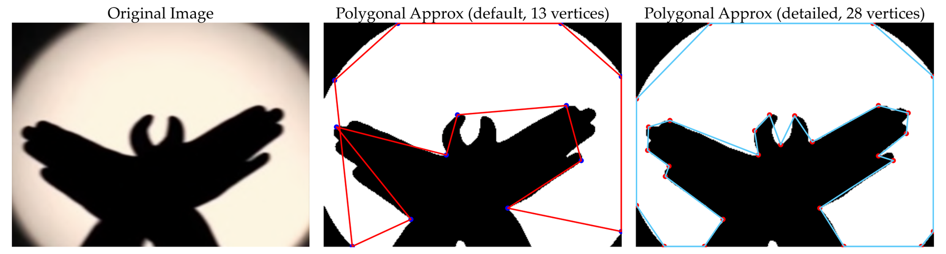


Figure 9. Comparison of polygonal approximations for a hand shadow puppet silhouette.

We also integrate ResNet34 with topological descriptors derived from skeletonized silhouettes, including branch/end-point counts and skeleton-to-area ratios.

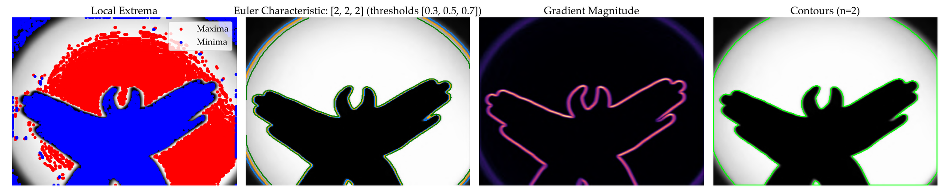


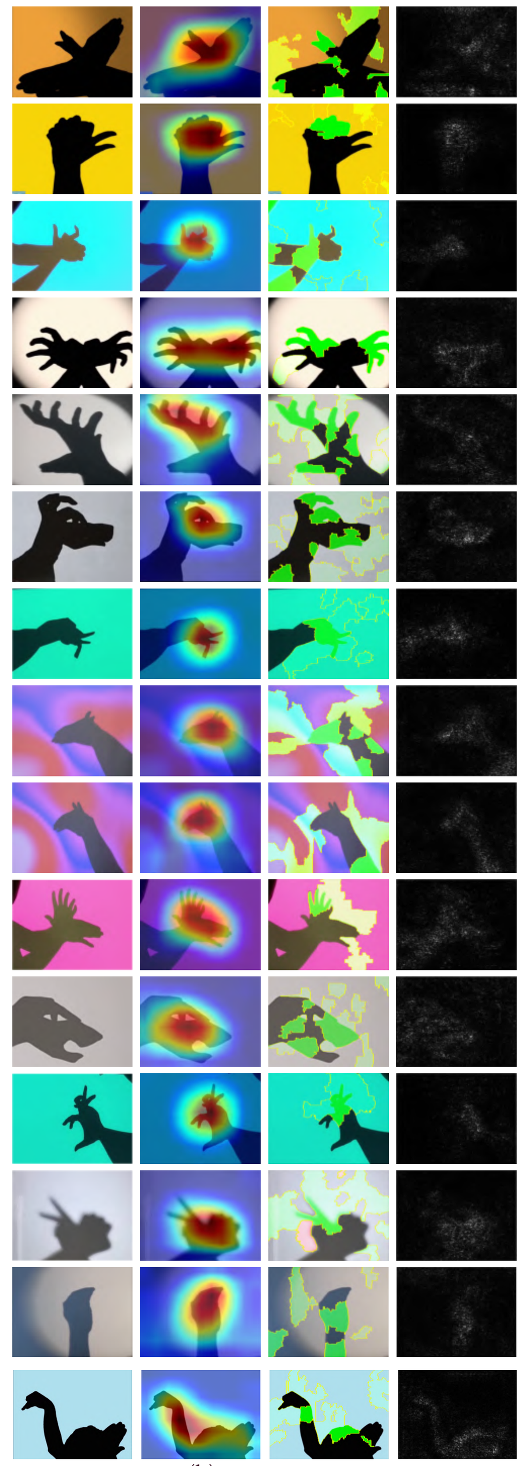
Figure 10. Topological features for a hand shadow puppet silhouette.

Results and Findings

The ResNet34 model yields the best performance with a top-1 accuracy of 94.97%. The vanilla version of the model also yields the highest top-2 accuracy, top-3 accuracy, Precision, Recall, and F1-scores of 97.23%, 98.23%, 0.9516, 0.9497, and 0.9491 respectively. We hypothesize that residual connections in ResNets help preserve low-level edge and contour information through identity mappings, ensuring that crucial silhouette boundaries aren't lost as the network deepens, making them better at capturing subtle variations.

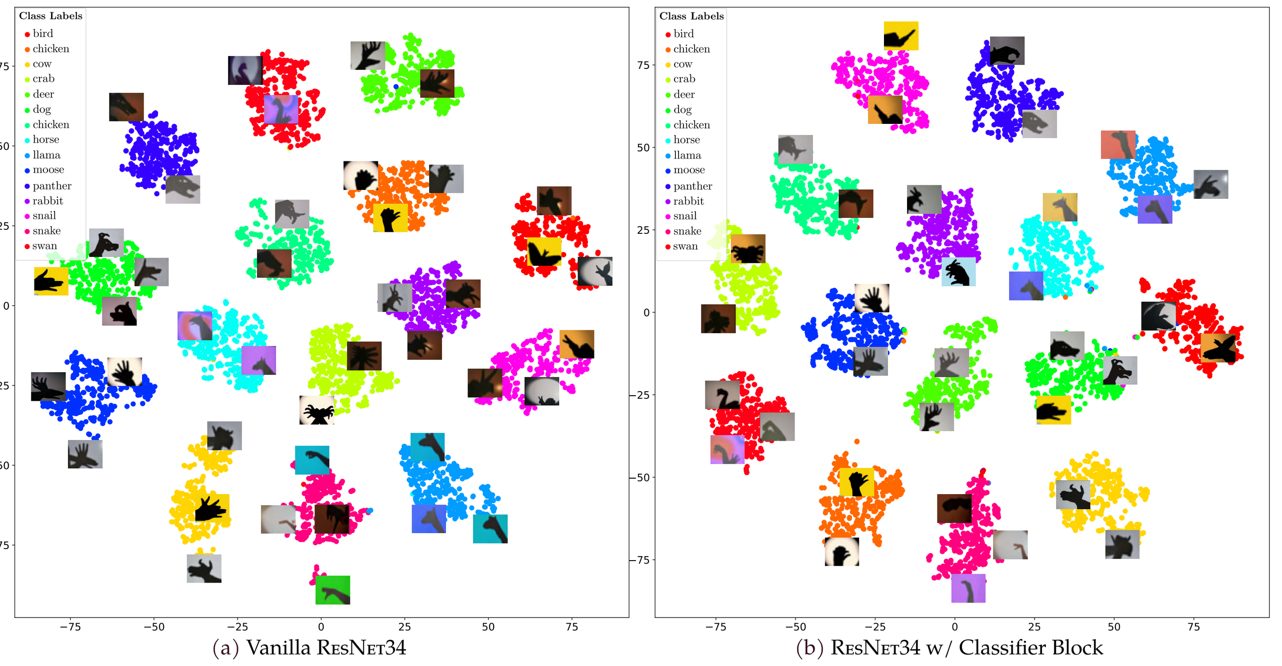
Models	Params.	Vanilla						w/ Classifier Block					
		Top-k Accuracy (%)			Precision	Recall	F1-score	Top-k Accuracy (%)			Precision	Recall	F1-score
		Top-1	Top-2	Top-3				Top-1	Top-2	Top-3			
ShuffleNetV2X10	2.3M	61.73	78.41	86.10	0.6559	0.6173	0.5970	88.73	93.98	96.10	0.8995	0.8873	0.8853
ViTB16	86.6M	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	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	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	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	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	87.8M	90.50	95.38	97.40	0.9128	0.9050	0.9042	90.20	95.40	97.08	0.9097	0.9020	0.9006
GoogLeNet	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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%)

Table 2. Performance comparison of the vanilla and modified versions of the image classification models.



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

Figure 11. The juxtaposition of original image samples from the HaSPeR dataset with their corresponding GradCAM Heatmaps, LIME Visualizations, and Saliency Maps (for the best-performing vanilla ResNet34 model).



(a) Vanilla ResNet34 (b) ResNet34 w/ Classifier Block

Figure 12 t-Distributed Stochastic Neighbor Embedding (t-SNE) feature representations of ResNet34

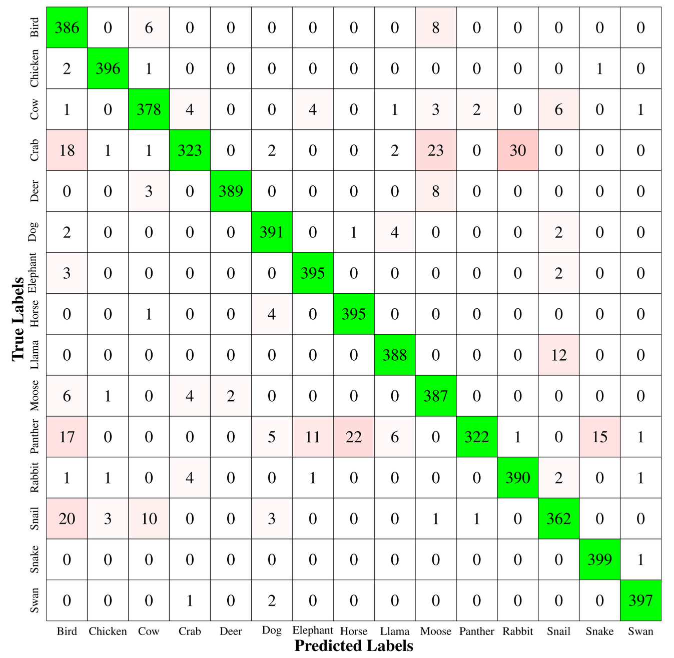


Figure 13. Confusion Matrix of vanilla ResNet34.

(a) Label: Crab, Predicted: Rabbit (b) Label: Moose, Predicted: Bird (c) Label: Panther, Predicted: Bird (d) Label: Llama, Predicted: Snail (e) Label: Snail, Predicted: Bird

(f) 'Rabbit' sample (g) 'Bird' sample (h) 'Bird' sample (i) 'Snail' sample (j) 'Bird' sample

Figure 14 Misclassified samples with visually similar samples of the predicted class.

Note

MobileNetV2, with only 3.5 million parameters, managed to surpass many of the other models in terms of performance.

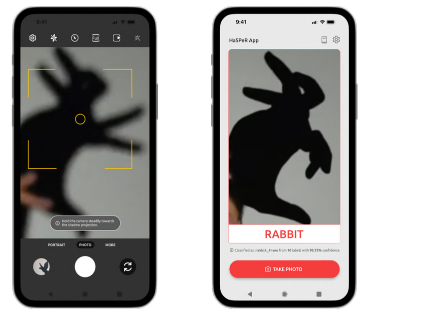


Figure 15. Android application for shadow puppet recognition.

References

[1] Syed Rifat Raiyan, Zibran Zarif Amio, and Sabbir Ahmed. HaSPeR: An image repository for hand shadow puppet recognition. *ICCV 2025 Workshop on Cultural Continuity of Artists*, 2025.