A. Related Work

The recognition and classification of hand shadow puppet images are intriguing problemspaces in the context of deep learning, albeit relatively underexplored. After rigorously analyzing the existing pool of research on the topic, we could identify several quasi-related works.

A.1. Image Classification and Recognition

Among the pioneering endeavors in hand shadow image classification was that of Huang *et al.* [22], who created SHADOW VISION—a system to emulate an immersive virtual shadow puppet theater experience, employing a user's hand gestures over an overhead projector to control the creation and manipulation of objects within a 3D Open Inventor[10] environment. The chain of stages underlying the implementation of SHADOW VISION were acquisition, segmentation, feature extraction, and recognition of the infrared shadow puppet images. They also adopted a 3-layer neural network and the centralized contour moments modeling technique, using 13 features (7 moments of the object, length, angle, and the 4 endpoints of the axis of inertia). The data used for this study isn't publicly available, and the methodology can be deemed somewhat obsolete in the modern purview, due to being supplanted by the emergence of deep learning models. Some recent works explore different convolutional models to assess their efficacy in Indonesian shadow puppet recognition. Sudiatmika and Dewi [55], Sudiatmika *et al.* [56] used the deep CNN models, ALEXNET [27] and VGG-16 [52], and constructed a dataset of 2,530 images spanning 6 classes of puppets from museums in Bali. They also experimented with other convolutional models, such as MASK R-CNN [16] and MOBILENET [19], in two separate studies [44, 57].

In a similar spirit, our work is an endeavor towards establishing a performance benchmark of the recent SOTA feature extractor models for hand shadow puppet contour images, in a more large-scale and comprehensive manner.

A.2. 3D Modeling and Human Motion Capture

One of the earliest works involving silhouettes is a study by Brand [5] that explored the mapping of monocular monochromatic 2D shadow image sequences of humans to animated 3D body poses, using a configural and dynamical manifold created from data with a topologically special hidden Markov model (HMM), acquired via the process of entropy minimization without resorting to any articulatory body model. Several advances in vision-based human motion capture and analysis since then have leveraged human silhouette templates [7, 42], more specifically, hand and finger silhouettes [65–67].

A.3. Robotics

Huang *et al.* [23] introduced computer vision-aided shadow puppetry with robotics by matching shape correspondences of input images. They claimed that due to the physical limitations of human arms, it is often not feasible to construct complex shadow forms. Instead, they developed a framework that enabled them to produce shadow images with the mechanical arms of a robot. The authors built a library of shadow images and used them to orient the robotic arms into a formation resembling the intended shadow puppet. The data used for this study isn't publicly available.

A.4. Human-Computer Interaction

The authors of [75] proposed a framework for controlling two Chinese shadow puppets—a human model and an animal model, with the use of body gestures via a Microsoft Kinect sensor. Carr and Brown [6] conducted a similar work by building a real-time Indonesian shadow puppet storytelling application that is capable of mimicking the full-body actions of the user, using the Microsoft Kinect sensor. In order to leverage contactless gesture recognition (CGR) to teach traditional Chinese shadow puppetry to beginners, Tsai and Lee [64] developed a system using Leap Motion sensors. These studies on digitizing the art of shadow puppetry, or puppetry in general, were influenced to some extent by other similar works in the gesture recognition domain [12, 13, 29, 30, 36, 71]. Tang *et al.* [63] developed an intelligent shadow play system, called SHADOWTOUCH, which includes a multidimensional somatosensory interaction module coupled with an automatic choreography module, to facilitate natural interaction between the shadow play figures and the human users.

The motif of our work tessellates well with the core objectives of the aforementioned research works. The utilization of digitized traditional arts serves as a means to preserve their inherent legacies, and HASPER can be a potent contribution to the contemporary pool of resources to facilitate such innovative digitization for ombromanie.

B. Experimental Setup: Additional Details

B.1. RESNET34 Architectural Enhancements

B.1.1. Silhouette Polygonization

We augment RESNET34 with handcrafted polygonal features extracted from the silhouette contours of hand shadow puppets, using Douglas-Peucker approximation [10] and geometric shape descriptors [74]. These features capture structural cues such as convexity, angularity, and polygonal regularity that complement the visual representations learned by the CNN feature extractor model. Hand shadow puppets inherently possess distinct geometric forms and outlines, which traditional CNNs might implicitly learn but

[10] Open Inventor™ toolkit — https://www.openinventor.com/

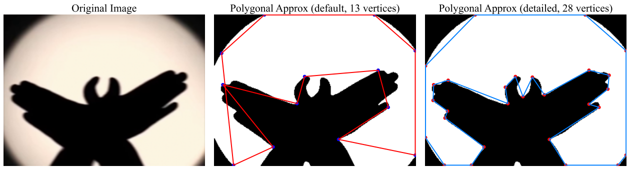

Original Image
Polygonal Approx (default, 13 vertices)
Polygonal Approx (detailed, 28 vertices)

Figure 13. Comparison of polygonal approximations for a hand shadow puppet silhouette.

not explicitly represent. To augment the classification process with this crucial geometric information, we test the integration of features derived from polygonal approximations and contour analysis.

1. **Silhouette Preprocessing:** The input image is first converted to grayscale and then binarized using Otsu's method [43]. Morphological closing and opening operations [51] are subsequently applied to remove small holes, smooth contours, and eliminate noise, ensuring a clean and consistent silhouette.
2. **Basic Contour Properties:** For the largest detected contour (representing the main puppet silhouette), fundamental geometric properties (area, perimeter, compactness, aspect ratio, solidity) are computed using OpenCV [25].
3. **Convexity Defects Analysis:** Hand shadow puppets often feature distinct *"finger-like"* protrusions and indentations. We analyze convexity defects (the regions between the contour and its convex hull) to quantify these characteristics. Features include the number of significant defects, as well as the mean, standard deviation, and maximum depth of these defects. A specific check is incorporated to identify *"finger-like"* defects based on their depth relative to the perimeter and the sharpness of the angle at the defect point.
4. **Hu Moments:** These are seven scale, rotation, and translation-invariant moments derived from the central moments of the contour [20]. Hu moments are powerful shape descriptors, capturing intrinsic characteristics regardless of the puppet's position, size, or orientation in the image. We use their logarithmically scaled values for better numerical stability.
5. **Ellipse Fitting:** An ellipse is fitted to the main contour, and the ratio of the contour's area to the fitted ellipse's area is computed. This metric provides insight into how well the silhouette can be approximated by an elliptical shape.
6. **Polygon Approximation Features:** The Douglas-Peucker algorithm (`cv2.approxPolyDP`) [10] is used to simplify the contour into a polygon with a reduced number of vertices, controlled by an epsilon (ϵ) factor (as evident in Fig. 13).
7. **Skeleton-Based Features:** A morphological skeleton of the binary silhouette is generated [28]. This skeleton represents the medial axis of the shape and provides valuable structural information.
   - *Skeleton Ratio:* The ratio of skeleton pixels to total silhouette pixels, indicating the *"thinness"* of the shape.
   - *Branch Points and Endpoints:* The counts of these critical points on the skeleton, which correspond to junctions and extremities (like fingertips) in the hand shadow puppet.

These features collectively form a robust and comprehensive polygonal descriptor vector, explicitly encoding geometric characteristics that are highly relevant for distinguishing between different hand shadow puppet forms.

B.1.2. Topological Features

This variant integrates RESNET34 with topological descriptors derived from skeletonized silhouettes, including branch/end-point counts and skeleton-to-area ratios [3, 28]. Such features model the internal articulation and connectivity of hand shapes, enabling finer discrimination of visually similar gesture classes. While CNNs excel at extracting hierarchical and abstract visual patterns, they may not explicitly capture the fundamental *"shape"* or connectivity of objects, which is crucial for silhouette-based recognition. To address this, we integrate topological features derived from persistent homology [11] into the classification pipeline. The topological feature extraction module is designed to quantify intrinsic shape properties invariant to continuous deformations, such as stretching or bending. This is particularly relevant for hand shadow puppets, where variations in hand posture can alter geometric appearance while preserving the underlying topological form.

1. **Betti Curves (Simplified Persistent Homology):** We approximate Betti numbers by analyzing the image at various filtration levels (thresholds from 0 to 1). β_0 is the number of connected components, reflecting the fragmentation or unity of the silhouette. β_1 is an estimation of the number of *"holes"* or loops within the silhouette. We derive this by considering the difference in pixel counts between the binary image and its morphologically filled counterpart, normalized to reduce sensitivity to small noise. These Betti curves provide a multiscale topological signature of the image.
2. **Critical Points Analysis:** We identify local maxima and minima within the smoothed grayscale image. The counts and densities (normalized by total pixels) of these critical points offer insights into the image's *"peaks"* and *"valleys,"* which correspond to salient features of the silhouette's shape.
3. **Morphological Features:** Standard morphological operations, opening and closing, are applied to the binary silhouette at various kernel sizes (3, 5, 7). The ratio of pixels in the opened/closed image to the original binary image's pixel count provides measures of the object's robustness to small protrusions/indentations and its overall

compactness.

4. **Euler Characteristic at Multiple Scales:** The Euler characteristic (χ = connected components $-$ holes) is a fundamental topological invariant. We compute this characteristic at different binarization thresholds (0.3, 0.5, 0.7) to capture how the global topology of the silhouette evolves across different levels of detail.
5. **Gradient-Based Features:** To capture edge information, Sobel filters are used to compute horizontal and vertical gradients. Statistical properties (mean, standard deviation, 90th percentile) of the gradient magnitude provide a summary of the image's edge strength and complexity.
6. **Contour-Based Features:** Utilizing OpenCV's contour detection, we extract features directly from the silhouette's boundaries. This includes the number of distinct contours, and the mean and standard deviation of their areas and perimeters. These features directly characterize the complexity and size of the hand shadow's outline.

These diverse topological features are concatenated into a single, fixed-dimension vector, designed to provide a comprehensive, invariant representation of the silhouette's inherent shape.

B.2. Performance Metrics

We use top-k validation accuracy values (with $k = 1, 2, 3$), Precision, Recall, and F1-score as evaluation metrics to perform comparative analyses among the aforementioned models. The latter three judgment criteria are used due to the slightly imbalanced nature of HASPER's professional source clips, as evident in Tab. 1.

B.3. Hyperparameters and Optimizer

We use Stochastic Gradient Descent (SGD) [26], with a learning rate $\alpha = 0.001$ and momentum $\gamma = 0.9$, as the optimizing method, and Cross Entropy Loss as the loss metric for all the models. To decay the learning rate, we use Step Scheduler, which decays α by 0.1 every 5 epochs. Each model undergoes training for 50 epochs to ensure equitable comparison, and we empirically ascertain that 50 epochs are sufficient for all of the models to achieve convergence.

B.4. Data Augmentation and Preprocessing

In order to generate a more diverse pool of training samples, we also incorporate data transformation techniques[11]—Random Resize, Random Perspective, Color Jitter, Random Invert, Random Horizontal Flip, Random Crop, Random Rotation, Gaussian Blur, and Random Affine with translation and shearing—while training the models. We choose these data augmentation techniques since the classes in HASPER are mostly rotationally asymmetric and incongruent. Consequently, the augmented samples aid in eliciting better generalization abilities and robustness for all the models. The input images that are fed to the models are appropriately resized *a priori* using Bicubic Interpolation.

C. Feature Space Visualization and Analysis

In order to visualize the learned feature space of RESNET34, we resort to the dimensionality reduction technique called t-Distributed Stochastic Neighbor Embedding [69] since it can preserve the proximity of high-dimensional data points. For high-dimensional data residing on or proximate to a low-dimensional, non-linear manifold, it becomes imperative to preserve this proximity of the collapsed low-dimensional representations for closely resembling data points. Achieving such proximity preservation is often unattainable through linear mappings such as Principal Component Analysis (PCA), which is why we opt for the t-SNE dimensionality reduction approach. We can pragmatically infer from the 2D-collapsed visualizations of the high-dimensional feature representations in Fig. 14, that the classes are nicely clustered and congealed with minimal overlaps and outliers. This enables the model to easily determine the decision surface in the high-dimensional feature space and perform very well on the classification task.

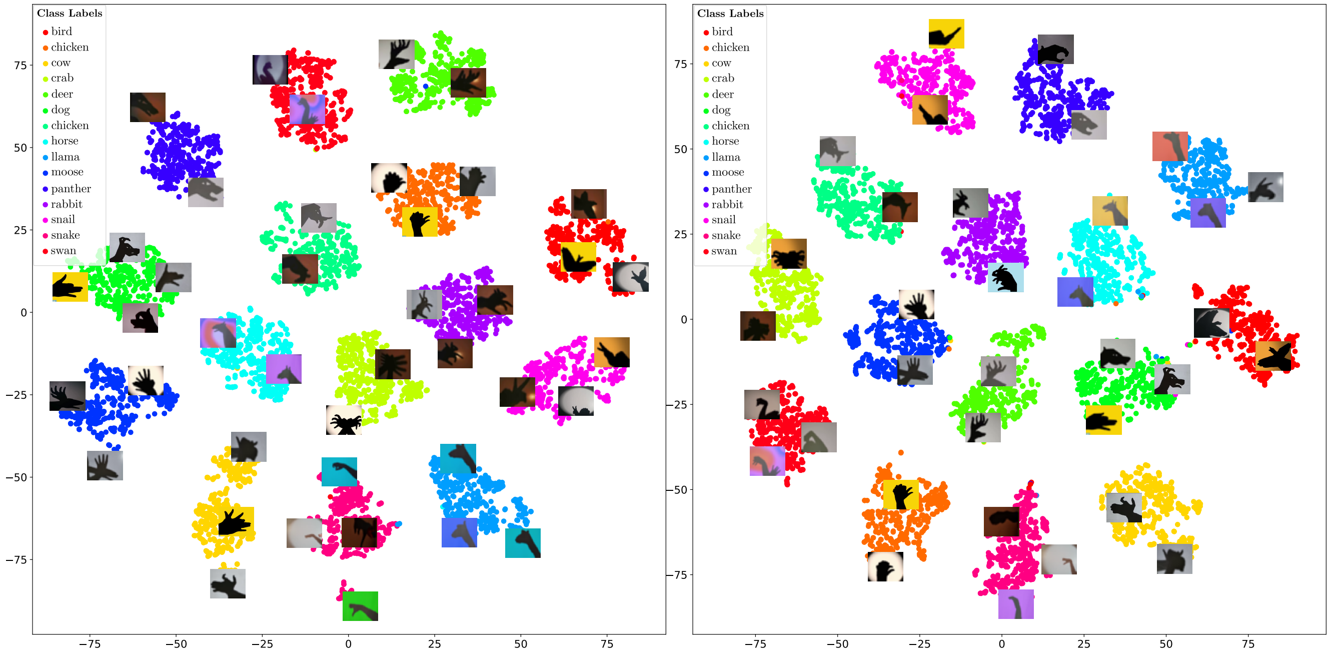
D. Additional Requirements for Teaching App

The system must operate with minimal computational overhead, ensure real-time responsiveness, and maintain low latency. This necessitates a pragmatic tradeoff between the FLOP count and classification accuracy, with the requisite model compression and optimization techniques. Given the potential variability in device camera capabilities, the application must have preprocessing steps including, but not limited to, denoising, adaptive contrast enhancement, and sharpening the input feed to mitigate artifacts. In recognition of the diverse motor capabilities of users, particularly younger learners and users with dexterity impairments, the application must have intelligent motion compensation to stabilize the shaky camera inputs. Adding to the desiderata is the consideration of a suitable UI/UX that is tailored for the pediatric user base, as is done in handwriting teaching apps for children [2], because an intuitive and enjoyable learning experience is of paramount importance for educational apps. This includes providing step-by-step tutorials, interactive guides with progressive difficulty, and illustrative diagrams to demonstrate the creation of hand shadow puppets.

E. Avenues of Improvement

To reduce the number of misclassifications, models need to be imbued with the ability to learn certain nuanced features.

[11] https://pytorch.org/vision/stable/transforms.html



(a) Vanilla ResNet34

(b) ResNet34 w/ Classifier Block

Figure 14. *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) feature representations of ResNet34.

In light of the contemporary image classification literature, we can opt to use Kolmogorov–Arnold Networks (KAN) [35] instead of a simple Multilayer Perceptron (MLP) as the classifier block. The use of Convolutional KANs [4] has yielded good results in many image classification benchmarks. The task of image classification on HaSPeR can be dubbed as a silhouette classification task, which is why we can leverage topological features of the shadow contours to achieve better results [31]. The silhouette polygonization algorithm (PoG), as proposed by Göçmen and Akata [14], may aid in achieving better classification accuracy. Other possible avenues may involve the use of ensemble methods coupled with voting schemes, or resorting to data augmentation with synthetically generated samples, but we defer the exploration of these hypotheses for future research endeavors.

F. Impact Statement

This research is an impetus towards utilizing AI tools to revitalize the hitherto underexplored cinematic art form of hand shadow puppetry. Such tools may help understand the creativity frontier in generative models, facilitate the development of applications to teach shadowgraphy, and unveil several prospects for entertainment. The existing works, though distally relevant to shadowgraphy, explore the digitization of such precursory art forms via approaches that have since been rendered primitive and obsolete. The novel dataset that we introduce in this paper, namely HaSPeR, consists of 15,000 diverse samples garnered from performance clips of variably skilled puppeteers. Our extensive benchmarking reveals that the task of classifying the puppet silhouettes is reasonably solvable using lightweight and convolutional feature extractor models, with accuracies of 94.97% by ResNet34 and 92.38% by MobileNetV2. HaSPeR, as a data resource for all intents and purposes, can be a potential stride towards systematically preserving this artistic practice.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The codes and datasets generated, written, and/or analyzed during the research project are available in the HaSPeR GitHub repository (https://github.com/Starscream-11813/HaSPeR) and in the HaSPeR Hugging Face repository (https://huggingface.co/datasets/Starscream-11813/HaSPeR).

Funding Sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. All associated costs were self-funded by the authors. Access to computing resources was provided by the Systems and Software Lab (SSL) of the Islamic University of Technology (IUT).