Abstracted Gaussian Prototypes for One-Shot Concept Learning

Chelsea Zou [1] **Kenneth J.Kurtz** [1]

Abstract

We introduce a cluster-based generative image segmentation framework for one-shot learning to encode higher-level representations of visual concepts using Gaussian Mixture Models (GMMs). The inferred parameters of each Gaussian component represents a distinct topological subpart of a visual concept. Sampling new data from an ensemble of these parameters generates augmented subparts to build a more robust prototype for each concept – what we propose as the *Abstracted Gaussian Prototype* (AGP). Using AGPs, our framework addresses both one-shot classification tasks through a cognitively inspired similarity metric and one-shot generative tasks through a novel AGP-VAE pipeline employing variational autoencoders (VAEs) to generate new class variants. This simple, standalone framework yields impressive classification accuracy, while also performing a breadth of conceptual generative tasks that most approaches do not attempt. Results from human judges reveal that our generative pipeline produces novel examples and classes of visual concepts that are broadly indistinguishable from those made by humans.

1. Introduction

The ability of humans to acquire novel concepts after only minimal exposure to examples is an important constituent of general intelligence. Humans have the remarkable ability to quickly abstract concepts and extrapolate from a few prior examples (Lake et al., 2015), thereby allowing for efficient and adaptable learning. On the contrary, most current machine learning (ML) architectures require large amounts of data to learn, massive numbers of parameters (e.g., GPT-3: 175B, AlexNet: 62.3M, VGG16: 138M), and in some cases pre-training or pre-established constraints on representation and processing (Hendrycks et al., 2019; Han et al., 2021; L'heureux et al., 2017; Zhou, 2016). Hence, a key computational challenge is to understand how an intelligent system with minimal complexity and without reliance on external supports can acquire new concepts from extremely restricted available training data (Chollet, 2019).

In this paper, we approach the challenges of one-shot learning (Wang et al., 2020; Kadam & Vaidya, 2020) by developing a framework that can perform both classification and generative tasks defined by the Omniglot challenge of handwritten characters, a testbed designed to study human and ML within the cognitive science and artificial intelligence communities (Lake et al., 2019). In the classification task, a single image of a novel character is presented and the aim is to correctly identify another instance of that character from a choice set of characters. In the generative tasks, the goal is to create new variants of characters that are indistinguishable from human drawings. While the classification task has received much attention, there has been limited success in the attempt to achieve both types of tasks from the same system (despite an emphasis on exactly this breadth of functionality in the Omniglot challenge).

To address both tasks, we propose the *Abstracted Gaussian Prototype* (AGP), which leverages Gaussian Mixture Models (GMMs) to flexibly model visual concepts of handwritten characters by capturing their stroke subparts as inferred Gaussian components. GMMs are unsupervised clustering algorithms that represent data as a finite sum of Gaussian distributions (Duda et al., 1973; Yu et al., 2015; Liang et al., 2022; Reynolds et al., 2009). A key insight driving our approach is viewing the pixels of an image as instances in the domain of an individual character. We use GMM-based clustering to model the topological subparts of each class in terms of its unique distribution. Based on the parameters underlying each component's distribution, the GMMs are then used to generate additional model-consistent pixels that augment the subpart representations. Finally, the collective ensemble of these generated subparts form what we refer to as the AGP, see Figure 1. AGPs provide a way to extrapolate beyond the constraints of a single available instance via the clustering functionality that imparts a quasi-structural analysis of the raw input into underlying parts with relative locations, and the generative functionality that enhances the item encoding by extrapolating its underlying model. This is a promising intermediate strategy between the problem of completely lacking a structural representation and

[1]Binghamton University (SUNY), NY, US. Correspondence to: Chelsea Zou <czou2@binghamton.edu>.

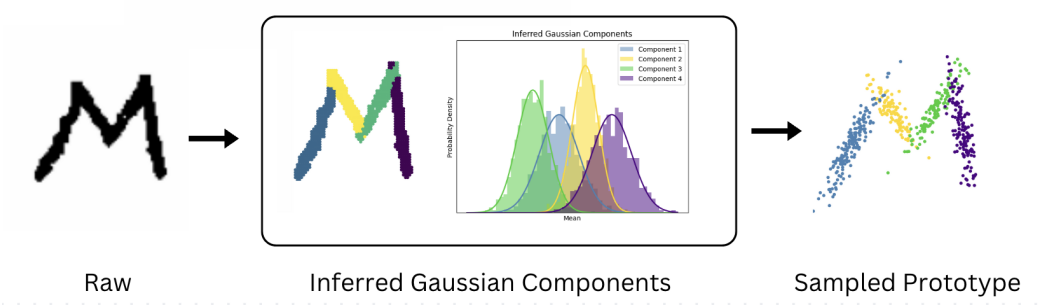


Figure 1. The raw image is shown on the left. The inferred clusters of the GMM are shown in the middle. Finally, the newly generated abstracted prototype is sampled from the inferred parameters.

that of being overly structured in the sense of requiring the overhead of a built-in fixed representational vocabulary of elements and relations, and often being compromised in terms of flexibility. In general, AGPs offer a method for generative image segmentation. The resulting higher-level representation (AGP) is successfully induced from a single example and supports a range of conceptual tasks. The GMM initially divides an image into different visual segments and this underlying model is used generatively to enhance item encodings or create new instances. AGPs are characterized by two elements: 1) probabilistic captures of data substructures, and 2) parameterized generative modeling. The first element allows the system to flexibly model different concepts and robustly handle noise. The second element allows the creation of diverse representations of each concept. These two features collectively comprise the benefits of using AGPs for both classification and generative tasks.

For one-shot classification, we leverage the generalized class representation of AGPs in combination with a cognitively-inspired metric to assess the similarity between AGPs. A classic and highly influential psychological theory of human similarity judgement is a set-theoretic approach known as the contrast model (Tversky, 1977). This similarity metric emphasizes an asymmetric weighting of the number of common and distinctive features between items being compared. In the context of the one-shot classification task, the features to be evaluated are pixel intersections between AGP images (further details below) and the classification decision is based on the highest similarity score to the target item.

For the challenging generative tasks in the Omniglot challenge, we build on the GMM-based approach outlined above to devise an AGP-VAE pipeline to create new classes. First, we synthetically generate a diverse training set of AGPs from each provided character. Then, we employ a variational autoencoder (VAE) (Kingma & Welling, 2013), a type of generative neural network architecture, to learn a continuous latent space that encapsulates a probabilistic distribution over different classes of generated AGP train sets. This novel AGP-VAE pipeline has the ability to interpolate between subparts of the discrete prototypes of the AGPs by sampling from a global feature space that encapsulates the local features of different classes. Instead of relying on the independent models of each class, we are able to represent the available classes in a subspace that can be sampled to produce novel character classes that conform to the distributional characteristics of provided data. Overall, our work provides several contributions:

1. We formulate the framework of an AGP as a generative image segmentation method that provides higher-level representations of each class.

2. We make novel use of a cognitively inspired similarity metric to perform one-shot classification tasks. We demonstrate the effectiveness of this simple approach compared to more computationally complex models.

3. We employ a novel AGP-VAE pipeline for generative tasks to create new variants of visual concepts. Results from human judges reveal that our model produces outputs that are broadly indistinguishable from human drawings.

Unlike existing approaches for one-shot learning, our framework provides a flexible and robust way to handle both classification and generative tasks. The system learns and performs these tasks without reliance on any foundation of pre-training or a built-in structural vocabulary. The approach is also highly transparent and relies on a small number of clear design principles that are novel applications of established computational constructs.

2. Prototypes in Human Concept Learning

Concepts are mental representations acquired through inductive learning processes and are used to categorize and

discriminate between different objects, events, and relational situations (Gregory, 2002; Goodman et al., 2008). Psychologists have emphasized that concepts are formed and represented through the integration and combination of simpler parts known as features or attributes (Schyns et al., 1998; Farhadi et al., 2009). In the concepts and categories literature in cognitive psychology, a prominent account is prototype theory in which mental representations of categories consist of storing the central tendency of feature values across observed members of a category (Rosch, 1973; Hampton, 2006; Posner & Keele, 1968; Minda & Smith, 2011). The prototype of a concept is a statistical average or an abstraction of all instances belonging to that category. Categorization is then understood as a matter of finding the best similarity match to stored prototypes in order to classify (Rosch & Lloyd, 1978; Mervis & Rosch, 1981). This approach has contrasted sharply with exemplar-based approaches that eschew abstraction in favor of simply storing labeled instances (Nosofsky, 1988; 1986). Prototype representations are economical and capture the intuition that the generic meaning underlying a category is represented explicitly and independently of its members.

In our framework, we adapt a form of prototype theory to address the Omniglot challenge in the ML literature. Computational approaches that operate by statistical learning are strongly linked to prototype theory (Biehl et al., 2016). For example, simple artificial neural networks (Rosenblatt, 1958) enact a functional version of prototype theory by adjusting the weights in either an auto-associative architecture trained on the members of a category or an MLP-style architecture trained to discriminate between classes (Ruck et al., 1990). Instead of explicitly storing prototypes, the weights are adapted to form an inductive model in which categories are generalized according to distance from their central tendency. In the case of one-shot learning, our approach transforms a single example into a prototype by taking a particular configuration of pixels as the basis for a set of probabilistic clusters that imply an underlying distribution. Instead of going from a collection of examples to a prototype, we propose going from a single example to a prototype which implies a collection. Specifically, our employment of GMMs captures the intricate variations and attributes inherent in visual concepts. The crux of our approach lies in the formation of an ensemble of augmented subparts achieved by using the parameters inferred from the Gaussian components of the GMMs. The AGP is an abstraction from a single case in order to create its own features (subparts) via clustering, establish the distributional central tendency and variability of its subparts, and inherently capture spatial relational properties between the subparts.

Just as the prototype theory of human categorization relies on similarity to the stored prototype in order to determine likelihood of category membership, we invoke a psychological similarity metric for classification relative to AGPs. Tversky's (1977) model of similarity proposed that individuals assess similarity by considering the number of featural differences and commonalities between items along with an additional design principle of highlighting (via greater weighting) the importance of the differences. The Tversky index is given by:

$$T(A,B) = \frac{|A \cap B|}{|A \cap B| + \alpha|A \backslash B| + \beta|B \backslash A|} \quad (1)$$

where $|A \cap B|$ is the size of the intersection of sets A and B, $|A \backslash B|$ is the size of the set difference of A without B, $|B \backslash A|$ is the size of the set difference of B without A, and α, β are non-negative parameters controlling the weight given to differences in the two sets. Following this, we use a simpler metric that has only one parameter to control the magnitude of penalty to the set differences comparing the intersections of AGPs, described in Section 5.1.2.

3. Related Works

One-Shot Classification: Many neural-based models have been successful at one or few-shot classification tasks (Finn et al., 2017; Santoro et al., 2016; Salakhutdinov et al., 2012). For instance, Siamese Neural Networks involve twin subnetworks sharing the same parameters that are trained to learn embeddings capturing the similarity or dissimilarity between pairs of instances (Koch et al., 2015; Chicco, 2021). Another approach through Prototypical Networks learn a representative prototype for each class based on the mean of the embeddings in the latent space and classify according to these distances (Snell et al., 2017). Similarly, Matching Networks work by employing an attention mechanism on embeddings of the labeled set of instances to forecast classes for unlabeled data (Vinyals et al., 2016). However, all of these neural-based classification approaches require an initial training phase for the network to learn a general understanding of the task. Furthermore, they are incapable of addressing generative tasks. Our proposed approach, on the other hand, offers a direct way for both classification and generative tasks to learn specific concepts from one, and only one, shot without background training on other data.

One-Shot Generation: One recent approach introduces GenDA for one-shot generative domain adaptation using pre-trained Generative Adversarial Networks (Yang et al., 2023; Goodfellow et al., 2014). GenDA designs an attribute classifier that guides the generator to optimally capture representative attributes from a single target image, in turn, synthesizing high-quality variant images. However, this approach relies on source models that are pre-trained on large-scale datasets such as FFHQ and Artistic-Faces dataset.

Bayesian Models: Significant progress has been made to address both one-shot classification and generative tasks

through Bayesian implementations, such as the Object Category Model (Fei-Fei et al., 2006) involving parametric representations of objects and prior knowledge when faced with minimal training examples. These Bayesian principles are manifested in the approach proposed by the original authors for the Omniglot dataset (Lake et al., 2011). In this system, a stroke model learns part-based representations from previous characters to help infer the sequence of latent strokes in new characters. An extension of this work introduces Bayesian Program Learning (BPL), which learns a dictionary of sub-strokes and probabilistically generates new characters by constructing them compositionally from constituent parts and their spatial relationships (Lake et al., 2015). BPL and the stroke model, however, requires the model to learn from stroke-data trajectories in order to extract and store a dictionary of primitive parses at the sub-stroke level. While this model first requires information from live-drawings, the approach may be unfeasible when temporally labeled sequential stroke data is not accessible. In contrast, our approach uses generative image segmentation to directly infer the sub-strokes of the characters using GMMs which allows our model to learn purely from raw, static images.

4. Background

In this section, we provide the mathematical background underlying GMMs and VAEs.

4.1. Gaussian Mixture Models

A GMM is a probabilistic clustering model that assumes the data is generated from a combination of multiple Gaussian distributions (Duda et al., 1973). Each Gaussian component $k \in K$ represents a cluster in the dataset, and is characterized by its unique parameters mean μ, standard deviation σ, and a weight π. A univariate Gaussian probability density function (PDF) for random variable X, which represents the probability of observing the given datapoint, is defined as the following:

$$P(X|\mu,\sigma) = \mathcal{N}(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(X-\mu)^2}{2\sigma^2}\right) \quad (2)$$

Similarly, a multivariate Gaussian PDF is defined as:

$$\mathcal{N}(\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right) \quad (3)$$

where X is now a d-dimensional feature vector, μ is a d-dimensional vector representing the means of the distribution, and Σ is a d x d covariance matrix. A finite GMM can be expressed by a weighted sum over K components:

$$P(X|\pi,\mu,\Sigma) = \sum_{k=1}^{K} \pi_k * \mathcal{N}(X|\mu_k,\Sigma_k) \quad (4)$$

satisfying the condition where:

$$\sum_{k=1}^{K} \pi_k = 1 \quad (5)$$

4.2. Variational Autoencoders

Variational autoencoders (VAEs) are neural networks characterized by an encoder-decoder architecture. Unlike traditional autoencoders (Bank et al., 2023), VAEs are equipped with a probabilistic framework based on variational inference. This framework employs networks to approximate otherwise computationally intractable posterior distributions, enabling the model to learn continuous representations of discrete input classes (Kingma & Welling, 2013; Kingma et al., 2019). This latent space facilitates the generation of diverse and novel samples, making VAEs a versatile tool for tasks such as image generation.

Encoder: The encoder of a VAE maps input data x to a latent space variable z, and is defined by an approximate posterior distribution:

$$q_\phi(z|x) = \mathcal{N}(z;\mu_z(x),\sigma_z(x)^2) \quad (6)$$

where $\mu_z(x)$ and $\sigma_z(x)$ are the mean and standard deviation of the approximate posterior learned by the encoder's weights and biases ϕ.

Sampling with Reparametrization Trick: To obtain a sample $z \sim q_\phi(z|x)$ from the latent space learned by the encoder, the reparametrization trick is used to maintain differentiability for backpropagation:

$$z = \mu_z(x) + \sigma_z(x) \odot \epsilon \quad (7)$$

where ϵ is typically sampled from a fixed standard normal distribution:

$$\epsilon \sim \mathcal{N}(0,1). \quad (8)$$

Decoder: The decoder of a VAE with parameters θ maps a sample z from the latent space back to the original feature space, generating a reconstruction $\hat{x}$ from the conditional likelihood distribution:

$$p_\theta(x|z) = \mathcal{N}(\hat{x};\mu_x(z),\sigma_x(z)^2) \quad (9)$$

where $\mu_x(z)$ and $\sigma_x(z)$ are the mean and standard deviation of the reconstructed data.

Loss Function: The training objective for VAEs is to maximize the Evidence Lower Bound (ELBO). The ELBO is defined by:

$$\mathcal{L}(\phi, \theta) = E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{10}$$

where the first term is the expected value of the log-likelihood of x given z and the second term is the Kullback-Leibler (KL) divergence between the approximate posterior and the prior distribution. The KL divergence measures the difference between two probability distributions and acts as a latent space regularization term that encourages the learned approximate posterior space to be close to the true posterior. Let J be the dimensionality of z. For the case of two Gaussian distributions, it is defined as:

$$D_{KL}(q_\phi(z|x)||p_\theta(z)) = \frac{1}{2}\sum_{i=1}^{J}\left(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1\right) \tag{11}$$

5. Approach

In this section, we describe and formalize our approach to the classification and generative tasks of one-shot learning in the Omniglot Challenge. The Omniglot dataset consists of 1623 hand-written characters taken from 50 different alphabets, with 20 examples for each class (Lake et al., 2019).

5.1. Classification Task

In a one-shot classification task, there is a set of N classes, denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$, and one available instance of each class. The given set of these single instances is denoted as $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Let $f : \mathcal{X} \rightarrow \mathcal{C}$ be the function that maps an instance to a corresponding class. Given an unseen instance q (usually referred to as the query) of an existing class, the goal of one-shot classification is to correctly determine the class of the query instance: $c_q = f(q)$, where $c_q \in \mathcal{C}$.

Our classification function is comprised of two steps: (1) AGP generation, and (2) the cognitively inspired similarity metric to determine classification choice. First, a separate GMM is used to model each concept, where each cluster of the model is assumed to represent a unique subpart of the concept. Next, we generate new abstracted subparts by probabilistically sampling new pixels from the inferred clusters. The collective ensemble of these generated subparts form the AGP, a higher level representation for each class, denoted $\mathcal{P}$. Using these prototypes, we use the similarity metric based on Tversky's (1977) contrast model to compute the match between the query prototype $\mathcal{P}_q$ and the prototype of each instance in the starting set $\mathcal{P}_i$ for $i \in \mathcal{X}$. The class which produces the highest similarity score is selected to be the class of q. The details of each step are formalized below.

5.1.1. Abstracted Gaussian Prototype (AGP) Generation

Each instance of a concept is provided as a binary image of pixels. Under the probabilistic framework of a GMM, let us define each sampled pixel as the realization of a random variable, characterized by its PDF corresponding to the inferred Gaussian component. Each instance in $\mathcal{X}$, along with q, is first segmented into its unique component subparts using a GMM, where $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$ represents the set of different subparts in each instance and k is a hyperparameter controlling the number of components. Here, $\mathcal{G}$ represents the mixture of Gaussian components of each instance, which allows the GMM to sample from the fitted distribution for each component g_i and generate new augmented subparts p_i. We define the ensemble of these subparts as the prototype $\mathcal{P}$ of the class, where $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$.

5.1.2. Similarity Metric

The similarity metric is computed between the query prototype $\mathcal{P}_q$ and each prototype of the available instances $\mathcal{P}_i$. Each prototype $\mathcal{P}$ is the entire set of its 2D image coordinates, generated by the sampled components in a GMM. To simplify the notation, let A be the set of all image coordinates for $\mathcal{P}_q$, and B be the set of all image coordinates for $\mathcal{P}_i$. We base our similarity metric on the Tversky index with the following equation:

$$S(A, B) = |A \cap B| - \beta\,|A \triangle B| \tag{12}$$

where $A \triangle B = (A \backslash B) \cup (B \backslash A)$ is the symmetric difference representing the non-intersections and $\beta > 1$ is a weight hyperparameter that ensures a larger penalty of this difference. The 2D image coordinates $a \in A$ and $b \in B$ will be denoted as (x_a, y_a) and (x_b, y_b), respectively. In order to calculate S by computing the number of intersecting pixels, we consider each pixel as a circle center with radius r. This ensures isotropy in our intersection calculation, maintaining consistent measurement of distances between pixels uniformly in every direction. Then, we can define:

$$|A \cap B| = \sum_{a\in A}\sum_{b\in B}\begin{cases}1 & \text{if } \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \leq r \\ 0 & \text{otherwise}\end{cases} \tag{13}$$

Using similar notation, it follows that the symmetric difference is:

$$|A \triangle B| = \sum_{a\in A}\sum_{b\in B}\begin{cases}1 & \text{if } \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} > r \\ 0 & \text{otherwise}\end{cases} \tag{14}$$

Finally, the $\mathcal{P}_i$ with the highest similarity score with $\mathcal{P}_q$ is

deemed to be of the same class.

$$c_q = f(q) = \arg\max_i S_i \tag{15}$$

If c_q matches the true class of the query instance, it is counted as a correct classification; otherwise, it is counted as an incorrect classification. To compute a more meaningful similarity score, we perform min-max scaling on each image pair and shift them to a center grid to align them as best as possible. For the similarity computations in the classification tasks, the query is shifted in eight different positions to test for the best alignment: up, down, left, right, and with 15 and 25 degrees clockwise and counterclockwise rotations. The hyperparameters of the circle radius, the density of pixels generated by the GMM, and the number of Gaussian components is optimized using grid-search based on classification accuracy in a trial of 100 iterations. We found the best performance to have radius = 1.6, density = 300, and number of components = 10.

5.2. Generative Tasks

The primary generative tasks in the Omniglot challenge are as follows:

1. Generating new exemplars of a particular class.
2. Generating new classes consistent with a particular alphabet given a starting set.
3. Generating entirely new classes (unconstrained).

For Task (1), only a single instance is used for the entire task. For Task (2), one instance per class from a given alphabet is allowed. Following (Lake et al., 2019), we use ten different instances. Similarly, Task (3) uses the same technique as Task (2) except that classes are randomly sampled across alphabets. Overall, the approach between these three tasks is similar and the only difference is in the starting instance(s) that are used.

There are three steps to generating new variations of exemplars from single instances. First, we synthetically increase the amount of training data by generating a larger number of AGPs per class. A key feature of this approach is that a range of prototypes with varied manifestations of abstraction can be generated by specifying a range of different number of components in the GMM for each class. This increases variation at the subparts level to diversify the training set. Second, we train a VAE across all the synthetic data to learn a continuous space of prototypes derived from the starting set. Lastly, we use a post-processing topological skeleton technique (Lee et al., 1994; Zhang, 1997) to refine the generated outputs. This layer denoises the reconstructed outputs of the VAE to ensure quality stroke images of the new class variants. The following sections describe each step in detail, and the pseudocode is shown below in Algorithm 1.

Algorithm 1 Generating New Variants

Input:
$\mathcal{X} \leftarrow$ set of single instances from N classes
$K \leftarrow$ range of Gaussian Components
GMM, VAE $\leftarrow$ trainable models
Skel $\leftarrow$ topological skeletonization function
Output:
$\Phi \leftarrow$ final training set of prototypes
$z \leftarrow$ reconstructed latent variables from trained VAE
$v \leftarrow$ final generated variant

for *i in N* **do**
  for *k in K* **do**
    train a $\text{GMM}_{i,k}(\text{x}_i)$ using k components
    $\mathcal{P} \leftarrow$ sample from $\text{GMM}_{i,k}$
    $\Phi_i \leftarrow \Phi_i \cup \{\mathcal{P}\}$ append $\mathcal{P}$ to set of class prototypes
  end
end
$\Phi \leftarrow \Phi \cup \{\Phi_i\}$ append class train set to final train set
train VAE(Φ)
$z \leftarrow$ sample and reconstruct VAE latent variables
$v \leftarrow$ Skel(z) postprocess reconstructed image

5.2.1. AGP Training Set

The approach from section 5.1.1 is used to generate more AGPs for each concept to synthetically increase the size of a training set $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_N\}$. This training set consists of a larger set of AGPs where $\Phi_i = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_D\}$, containing D new variants for each class, specified with a different value of k components. An equal number of variants is generated for each value of k, so that there are D/K variants for $k \in K$. In our pipeline, we generate D = 500 AGPs per class with K = {6,7,8,9 10} components.

5.2.2. VAE Interpolation

After generating the prototype training sets Φ_i for each class $i \in N$, the next step is to create continuous variations amongst these prototypes. To accomplish this, a VAE is trained across $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_N\}$ which is the enumeration of all prototype training sets, to learn a latent space representation that captures the underlying structures of these abstracted prototypes. The latent variables z are sampled accordingly to encourage semantic mixing between prototypes which are then decoded into the reconstructed variant images. For our generated outputs, we specify a convolutional VAE with the following details. The encoder of our model consists of two convolutional layers: a 32-filter 3x3 convolution followed by a 64-filter 3x3 convolution, both with a stride of 2 and ReLU activation. The decoder reshapes the latent vector to a 7x7x32 tensor, upsampled using two transposed convolutional layers with 64 and 32

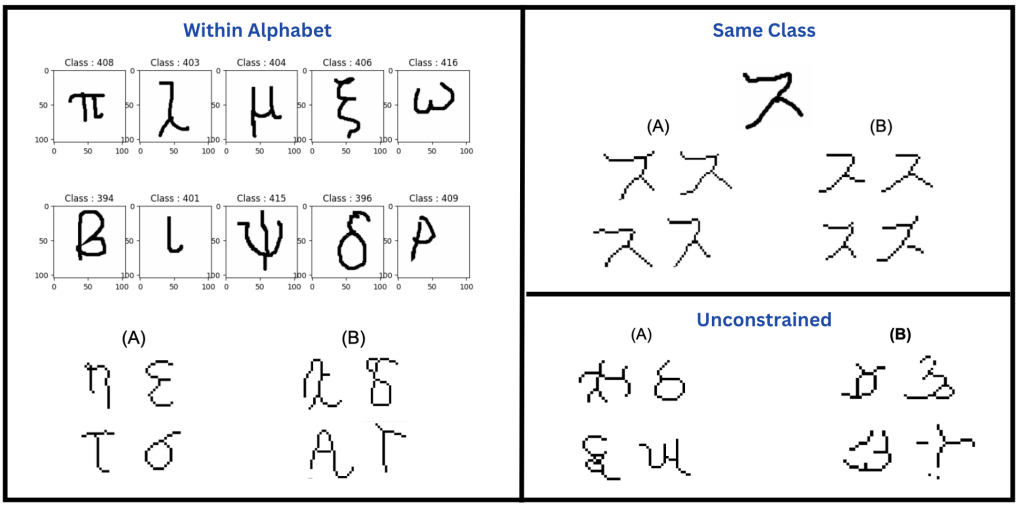


Figure 2. Visual Turing tests of the output characters generated from our AGP-VAE pipeline. The set of characters drawn by our model is B for within alphabet, A for same class, and B for unconstrained.

filters respectively, both with 3x3 kernels, a stride of 2, and ReLU activation. The final layer is a transposed convolution with a single filter, a 3x3 kernel, and stride of 1, outputting the reconstructed image. For training, the model uses the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1e-4. The loss function is computed as a combination of binary cross-entropy and the KL divergence between the learned latent distribution and the prior distribution. Finally, we train the model using a batch size of 32 for 50 epochs.

5.2.3. Topological Skeleton Refinement

The final step in this pipeline is a post-processing technique based on the work of (Lee et al., 1994; Zhang, 1997) on topological skeletons. Skeletonization is used often in image processing and computer vision to reduce the thickness of binary objects to one-pixel-wide representations while preserving the topological properties of objects. This step refines the reconstructed output images generated by the VAE and emphasizes the stroke-like properties of Omniglot characters. After each reconstructed image from the VAE is skeletonized, the final result is a collection of generated variants of characters for each task.

6. Results

6.1. Classification Tasks

The accuracy of our approach is evaluated based on the number of correct classifications averaged over 1000 trials in both 5-way and 20-way one-shot tasks. We test this in the context of unconstrained classes independent of alphabets, as well as a more challenging within-alphabet classification task. We compared our one-shot classification approach using the Tversky inspired similarity metric to a baseline euclidean distance metric calculated by the standard mean squared error. Results are summarized in Table 1. Additionally, we show how the simple AGP approach compared favorably relative to existing approaches in Table 2 adapted from (Koch et al., 2015).

	Euclidean Distance	**Our Similarity Metric**
5-Way Within	31.2%	**86.6%**
20-Way Within	8.9%	**71.0%**
5-Way Unconstrained	34.5%	**95.1%**
20-Way Unconstrained	19.3%	**84.2%**

Table 1. One-Shot Classification Scores Using Euclidean Distance

Model	20-Way Within
Nearest Neighbor	21.7%
Stroke Model	35.2%
Siamese Neural Network	58.3%
Deep Boltzmann Machine	62.0%
Hierarchical Deep	65.2%
AGPs	**71.0%**
Bayesian Program Learning	95.2%

Table 2. 20-Way Within One-Shot Classification Results on Omniglot

6.2. Generative Tasks

A "visual Turing test", as described in (Lake et al., 2013) is used to assess the quality of the generative outputs of the model. In this test, a set of characters produced by a human is displayed next to a set produced by the model. Human judges then try to identify which set was drawn by a human, and which set was generated by the model, see Figure 2 and

	Identification Accuracy	Preference
Mean	52.25%	55.25%
SD	8.13%	8.35%
Min	40.00%	43.00%
Max	63.00%	70.00%

Table 3. Descriptives for Average Scores Across Judges

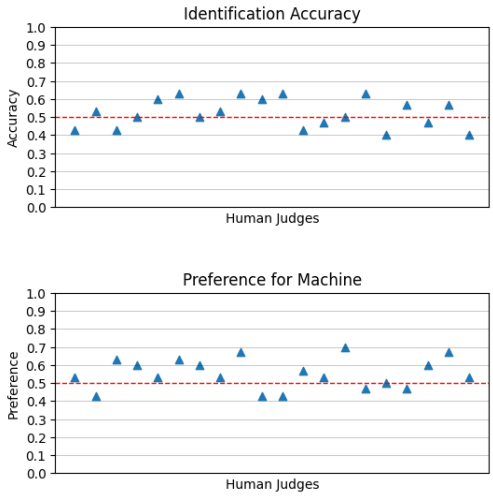


Figure 3. Each marker reveals the evaluation scores of a human judge averaged across all 30 sets of comparisons. The ideal performance of 50 percent is indicated by the dashed red line.

Appendix A. Our generative approach is evaluated based on the identification accuracy of the 20 human judges recruited online. The ideal performance is 50 percent, indicating that the judges cannot distinguish between characters produced by the human and the model, and the worst-case performance is 100 percent. Ten question sets with four instances from the human and four instances from our model were created for each of the three tasks (total of 30 sets). Additionally, we asked follow up questions after displaying each set of images to probe whether the machine's outputs could potentially surpass the quality of human generated characters. These questions were phrased as the following: (1) "Which set represents a better job of making four new examples of the given character?", (2) "Which set represents a better job of making four new characters that fit the given alphabet?", and (3) "Which set represents a better job of creating four new characters?"

Generating New Exemplars For the set of images corresponding to generating new exemplars of a particular class, the average identification accuracy across judges was (M = 55.50%, SD = 19.05%. Min = 30.00%, Max = 90%). The preference for machine-made in this specific task was (M = 34.50%, SD = 17.01%, Min = 10.00%, Max = 60.00%).

Generating New Concepts from Type For the evaluation of generating new characters belonging to an alphabet, the identification accuracy across judges was (M = 52.00%, SD = 14.73%. Min = 30.00%, Max = 90%). The preference for the machine-made was (M = 49.00%, SD = 15.18%, Min = 20.00%, Max = 80.00%).

Generating New Concepts (Unconstrained) For the final task of generating entirely new concepts independent of alphabet, the identification accuracy across judges was (M = 53.50%, SD = 18.99%. Min = 20.00%, Max = 100%). The preference for the machine-made in this task was (M = 51.50%, SD = 12.26%, Min = 30.00%, Max = 80.00%).

Overall Results for Generative Tasks The overall identification accuracy and preference scores averaged across all tasks and judges are shown in Table 3. In addition, Figure 3 provides a breakdown to reveal the subjective evaluations of each individual judge. These scores reveal promising results as identification accuracy is close to random chance. Notably, preference for machine-generated characters was slightly higher than human-drawn characters which could merit further exploration in terms of AI-generated content.

7. Conclusion

We present a novel approach for addressing the challenging problem of one-shot learning using AGPs. AGPs leverage GMMs to build representative prototypes for each concept by abstracting upon the subparts of the single available instances for each class. First, we proposed the AGP framework as a generative image segmentation method, offering a simple yet powerful method for encoding higher level representations of concepts from minimal data. Second, we introduced the novel use of a cognitively inspired similarity metric for classification tasks. Lastly, we developed a novel AGP-VAE pipeline employing VAEs to utilize AGPs for generating diverse and creative data variants close to indistinguishable from human drawings.

We aim to present AGPs as a novel approach that achieves high performance on the classification task and proves impressively adaptable to the generative tasks. Critically, this degree and breadth of success is achieved without high model complexity, without slow and demanding computation, without loss of transparency, without the need for external pre-training, and without invoking a complex symbol system for compositional structural recoding. We would highlight three takeaways from this research: (1) the deep value of the computational cognition framework for productive crosstalk between cognitive science and ML, (2) the potential for approaches that are intermediate in nature between the poles of the inductive/statistical and symbolic/algorithmic frameworks, and (3) the future potential of the design principles underlying the abstracted Gaussian prototype and the AGP-VAE pipeline in ML applications.

References

Bank, D., Koenigstein, N., and Giryes, R. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

Biehl, M., Hammer, B., and Villmann, T. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.

Chicco, D. Siamese neural networks: An overview. *Artificial neural networks*, pp. 73–94, 2021.

Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Dale, R. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

Duda, R. O., Hart, P. E., et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 1778–1785. IEEE, 2009.

Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., and Griffiths, T. L. A rational analysis of rule-based concept learning. *Cognitive science*, 32(1):108–154, 2008.

Gregory, M. The big book of concepts, 2002.

Hampton, J. A. Concepts as prototypes. *Psychology of learning and motivation*, 46:79–113, 2006.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019.

Kadam, S. and Vaidya, V. Review and analysis of zero, one and few shot learning approaches. In *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1*, pp. 100–112. Springer, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Welling, M., et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Lake, B. M., Salakhutdinov, R. R., and Tenenbaum, J. One-shot learning by inverting a compositional causal process. *Advances in neural information processing systems*, 26, 2013.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

Lee, T.-C., Kashyap, R. L., and Chu, C.-N. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

Liang, C., Wang, W., Miao, J., and Yang, Y. Gmmseg: Gaussian mixture based generative semantic segmentation models. *Advances in Neural Information Processing Systems*, 35:31360–31375, 2022.

L'heureux, A., Grolinger, K., Elyamany, H. F., and Capretz, M. A. Machine learning with big data: Challenges and approaches. *Ieee Access*, 5:7776–7797, 2017.

Mervis, C. B. and Rosch, E. Categorization of natural objects. *Annual review of psychology*, 32(1):89–115, 1981.

Minda, J. P. and Smith, J. D. Prototype models of categorization: Basic formulation, predictions, and limitations. *Formal approaches in categorization*, pp. 40–64, 2011.

Nosofsky, R. M. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

Nosofsky, R. M. Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: learning, memory, and cognition*, 14(4):700, 1988.

Posner, M. I. and Keele, S. W. On the genesis of abstract ideas. *Journal of experimental psychology*, 77(3p1):353, 1968.

Reynolds, D. A. et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

Rosch, E. and Lloyd, B. B. Principles of categorization. 1978.

Rosch, E. H. Natural categories. *Cognitive psychology*, 4 (3):328–350, 1973.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Ruck, D. W., Rogers, S. K., and Kabrisky, M. Feature selection using a multilayer perceptron. *Journal of neural network computing*, 2(2):40–48, 1990.

Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A. Learning with hierarchical-deep models. *IEEE transactions on pattern analysis and machine intelligence*, 35 (8):1958–1971, 2012.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Schyns, P. G., Goldstone, R. L., and Thibaut, J.-P. The development of features in object concepts. *Behavioral and brain Sciences*, 21(1):1–17, 1998.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Tversky, A. Features of similarity. *Psychological review*, 84(4):327, 1977.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Yang, C., Shen, Y., Zhang, Z., Xu, Y., Zhu, J., Wu, Z., and Zhou, B. One-shot generative domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7733–7742, 2023.

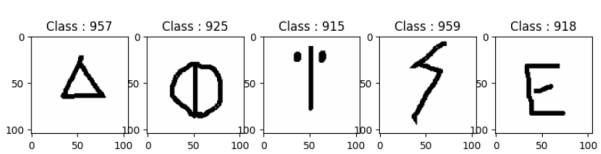
Yu, D., Deng, L., Yu, D., and Deng, L. Gaussian mixture models. *Automatic Speech Recognition: A Deep Learning Approach*, pp. 13–21, 2015.

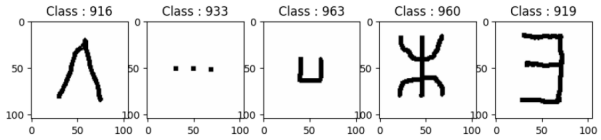
Zhang, T. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):337–343, 1997.

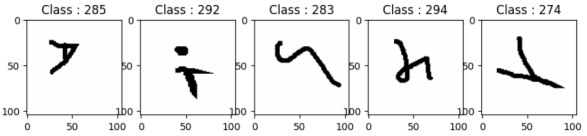
Zhou, Z.-H. Learnware: on the future of machine learning. *Frontiers Comput. Sci.*, 10(4):589–590, 2016.

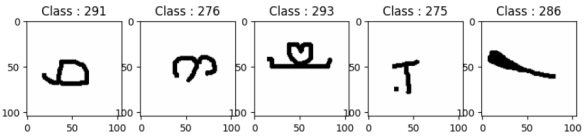
A. Appendix

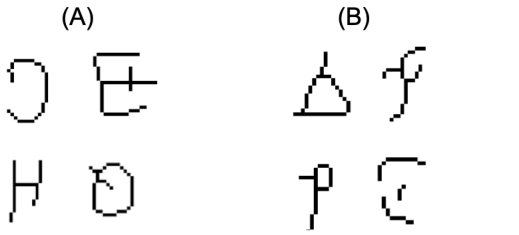
The full set of questions from our "visual Turing tests" given to the human judges. Character outputs from our AGP-VAE pipeline model (from left to right, top to bottom) are:
Within Class: A A B A B B A B B A
Same Class: B A A A B A B B A A
Unconstrained: A B B A A B A B A A

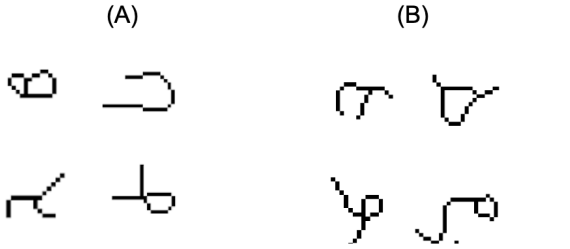

Class : 957
Class : 925
Class : 915
Class : 959
Class : 918

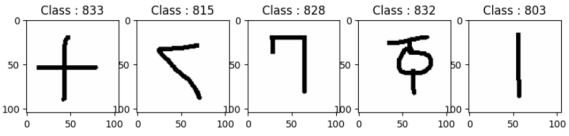

Class : 916
Class : 933
Class : 963
Class : 960
Class : 919

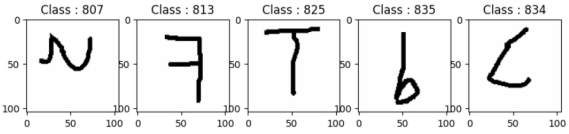

Class : 285
Class : 292
Class : 283
Class : 294
Class : 274

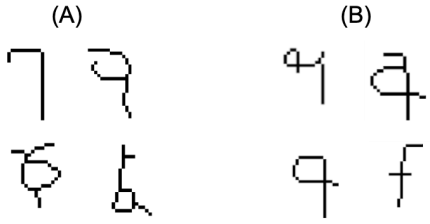

Class : 291
Class : 276
Class : 293
Class : 275
Class : 286

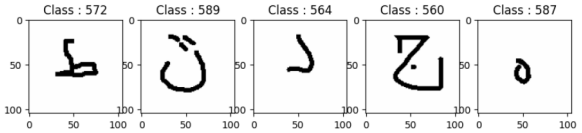

(A)
(B)

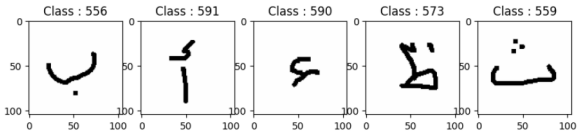

(A)
(B)

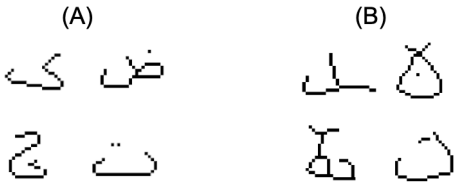

Class : 833
Class : 815
Class : 828
Class : 832
Class : 803

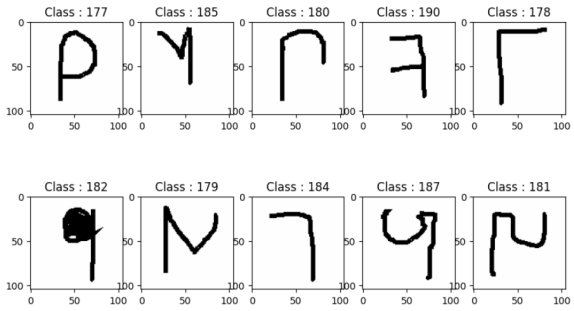

Class : 807
Class : 813
Class : 825
Class : 835
Class : 834

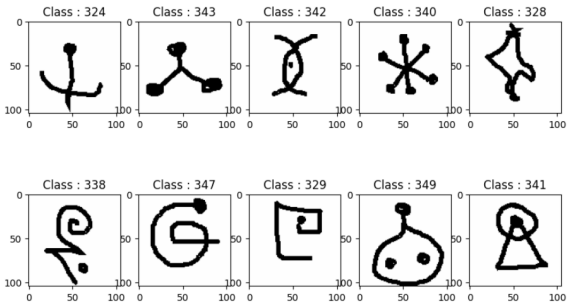

(A)
(B)

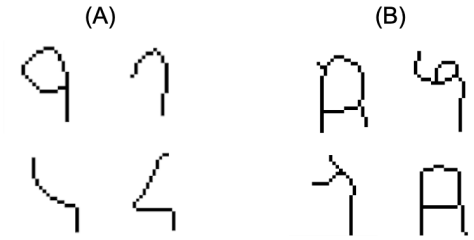

Class : 572
Class : 589
Class : 564
Class : 560
Class : 587

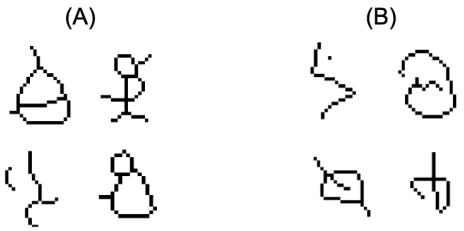

Class : 556
Class : 591
Class : 590
Class : 573
Class : 559

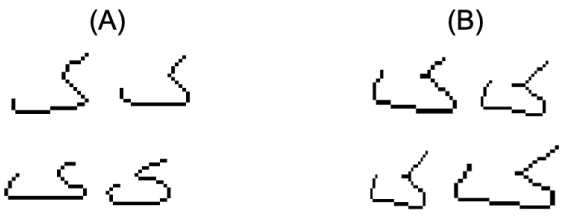

(A)
(B)

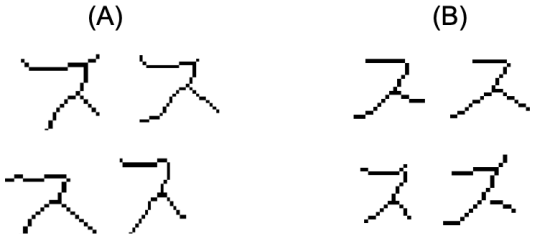

Class : 177
Class : 185
Class : 180
Class : 190
Class : 178
Class : 182
Class : 179
Class : 184
Class : 187
Class : 181


Class : 324
Class : 343
Class : 342
Class : 340
Class : 328
Class : 338
Class : 347
Class : 329
Class : 349
Class : 341

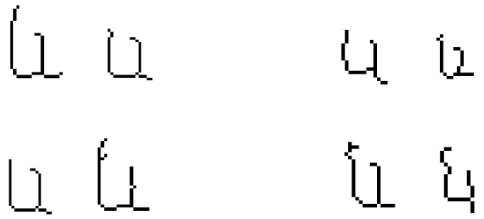

(A)
(B)


(A)
(B)


(A)
(B)

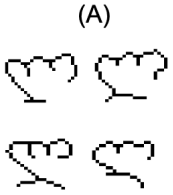

(A)
(B)

(A) (B)



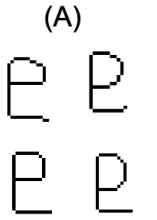
(A) (B)

(A) (B)

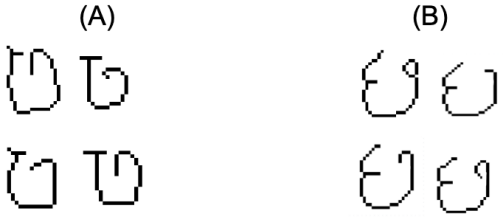


(A) (B)

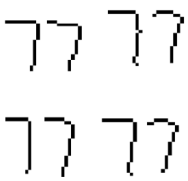
(A) (B)



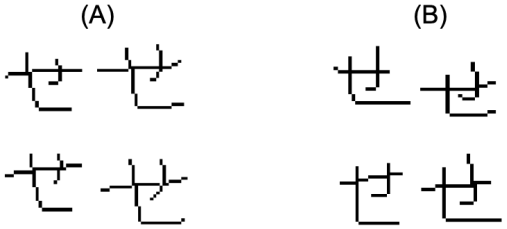
(A) (B)



(A) (B)

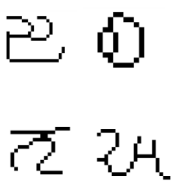


(A) (B)



(A) (B)

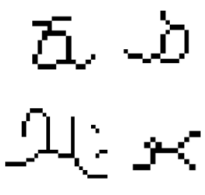
(A) **(B)**



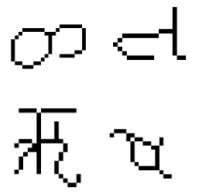
(A)

(B)

(A)

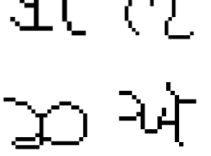


(B)

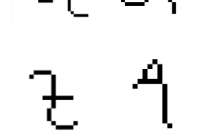


(A)

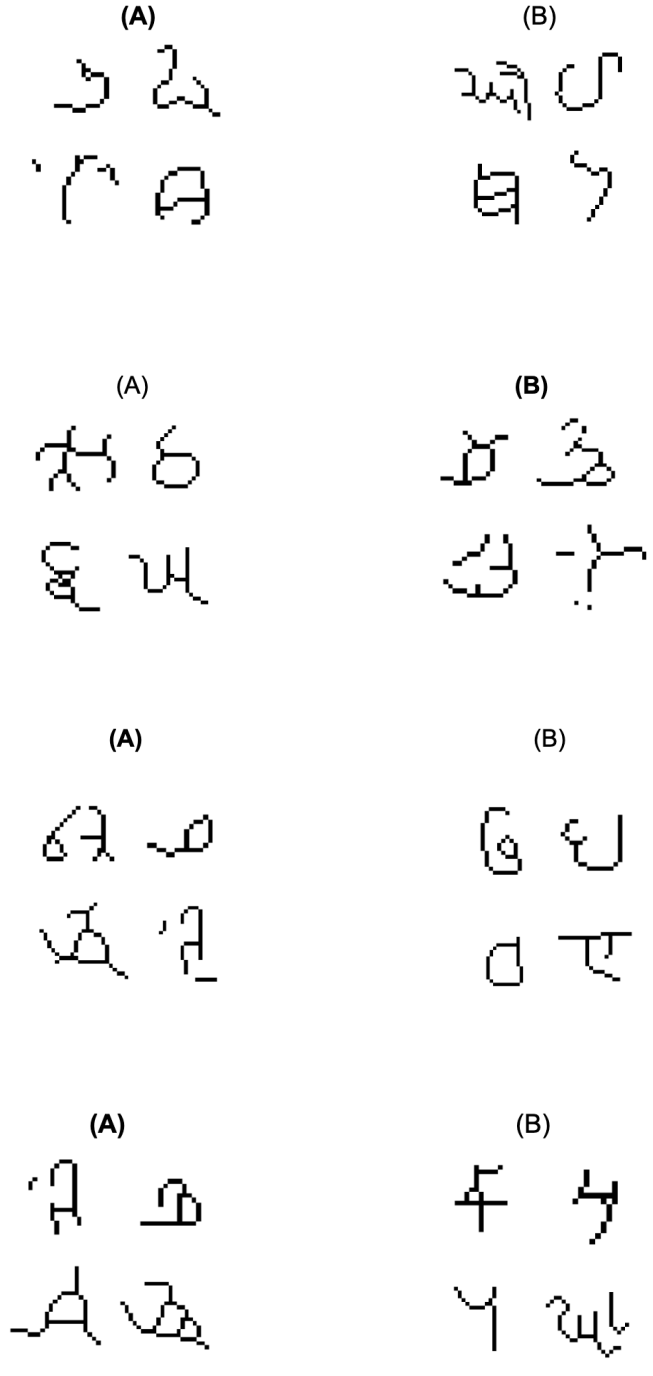
(B)



(A)



(B)


(A)
(B)
(A)
(B)
(A)
(B)
(A)
(B)