machine learning & knowledge extraction

Article

Digit Recognition Based on Specialization, Decomposition and Holistic Processing

Michael Joseph and Khaled Elleithy *

Computer Science and Engineering Department, University of Bridgeport, Bridgeport, CT 06604, USA; micjosep@my.bridgeport.edu

* Correspondence: elleithy@bridgeport.edu

Received: 12 July 2020; Accepted: 13 August 2020; Published: 18 August 2020

Abstract: With the introduction of the Convolutional Neural Network (CNN) and other classical algorithms, facial and object recognition have made significant progress. However, in a situation where there are few label examples or the environment is not ideal, such as lighting conditions, orientations, and so on, performance is disappointing. Various methods, such as data augmentation and image registration, have been used in an effort to improve accuracy; nonetheless, performance remains far from human efficiency. Advancement in cognitive science has provided us with valuable insight into how humans achieve high accuracy in identifying and discriminating between different faces and objects. These researches help us understand how the brain uses the features in the face to form a holistic representation and subsequently uses it to discriminate between faces. Our objective and contribution in this paper is to introduce a computational model that leverages these techniques, being used by our brain, to improve robustness and recognition accuracy. The hypothesis is that the biological model, our brain, achieves such high efficiency in face recognition because it is using a two-step process. We therefore postulate that, in the case of a handwritten digit, it will be easier for a learning model to learn invariant features and to generate a holistic representation than to perform classification. The model uses a variational autoencoder to generate holistic representation of handwritten digits and a Neural Network(NN) to classify them. The results obtained in this research show the effectiveness of decomposing the recognition tasks into two specialize sub-tasks, a generator, and a classifier.

Keywords: decomposition; digit recognition; face inversion effect; holistic processing; invariant features; knowledge representation; semi-unsupervised learning; specialization; template-based recognition

1. Introduction

Recognition is a challenging but essential task with applications ranging from security to robotics. To simplify face recognition, our brain process recognition in three phases. First, face detection, which is based on features (eyes, nose, and mouth) that distinguishes faces from other objects, this phase is often referred to as first-order information. Second, the facial features are integrated into a perceptual whole to form a holistic representation of the face. Third, facial variances that exist between individuals, such as the distance between the eyes, also referred to as second-order information, are extracted from the holistic representation to perform face discrimination [1]. Moreover, results from multiple face inversion experiments suggest that our brain processes a face in a specific orientation, the upright position [1–3]. Face inversion experiments on infants suggest that, by the age of one, infants develop familiarity with faces in the upright position and thus become more susceptible to experience the face inversion effect [4,5]. Another interesting observation is that, whenever a familiar face, with distorted second-order information, is presented upside down, the holistic representation is recalled from memory, and what is perceived is an undistorted face [1,6]. This implies that our brain uses critical information from the input image to recall from memory a holistic representation of the person's face. Based on those findings, we created a digit recognition model that uses Variational Autoencoder (VAE) that takes different representations and orientations of a digit image as input and project the holistic representation of that digit in a specific orientation. The hypothesis is that it will be easier for a learning model to learn invariant features and to generate a holistic representation than to perform classification. To verify this hypothesis, we conducted experiments on a specific class of objects, a handwritten digit. Using these experiments, we were able to empirically prove the hypothesis.

2. Materials and Methods

Our ability to distinguish different faces with high accuracy is extraordinary given the fact that all faces possess similar features, eyes, nose, and mouth, and the variance between faces is subtle. We achieve this level of accuracy by using a visual processing system that relies on specialization and decomposition.

Specialization: Several experiments using functional Magnetic Resonance Imaging (fMRI) and the face inversion effect prove that there is a particular area in the brain that responds only to face stimuli and in a specific orientation, the upright position. This was discovered in experiments conducted by Yin in 1969. He found a significant reduction in brain performance in face recognition when the incoming face stimulus is upside-down, [1–3], and the impairment was not as severe for objects [7,8]. This is evidence of a distinct neurologically localized module for perceiving faces in the upright position, which is different than the general visual pattern perception mechanisms use to identifying and recognizing objects. The fact that face inversion effect is absent in babies suggests that the brain specialization to process upright faces is learned over time, from experience [4,5].

Decomposition: Faces contain three types of information: First, the individual features such as eyes, nose, mouth, all of which have specific sizes and colors. Second, the spatial relations between those features, such as eyes above a nose and their relationship to the mouth, constitute what is called first-order or featural information. Third, the variance of those features concerning an average face or variation between the faces of different individuals constitutes the second-order or configural information [1,4,9]. Using decomposition, our brain sub-divides the complex task of recognition into three distinct sub-tasks: face detection, holistic representation, and face recognition [10]. Individual and local features are used for face detection [10], whereas configural information is used to retrieve from memory a holistic representation of which individual to identify [9,11].

"Holistic" representation is defined as "the simultaneous perception of the multiple features of an individual face, which are integrated into a single global representation" [12]. Therefore, it is an internal representation that acts as a template for the whole face, where the facial features are not represented explicitly but are instead integrated as a whole [8,11,13]. There is not a consensus on the content and structure of that holistic representation and conflicting conclusions on its contribution to the inversion effect [12]. Some believe that "critical spatial relationship is represented in memory" [9], while others believe both "configural information as well as the local information" are encoded and stored in memory [14]. There is, however, consensus on the brain's ability to perceive and process faces as a coherent whole. There are two well-known experiments that provide evidence for holistic processing: the composite effect and the part-whole effect [10,11]. In composite effect experiments, it is shown that it is difficult for subjects to recognize that two identical top halves faces are the same when they are paired with different bottom halves [5,15]. In part-whole effect experiments, subjects are observed having difficulty recognizing familiar faces from isolated features [13].

In another experiment (See Figure 1), the configural information in the faces are altered to the point of being grotesque (eyes are placed closer to each other, shorter mouth to nose spatial relation, etc.). When those altered faces were presented in an inverted orientation, the "distinctiveness impressions caused by distorted configural information disappeared" [1,6]. The brain recalls from memory, holistic representation of the face that is free of distortion.

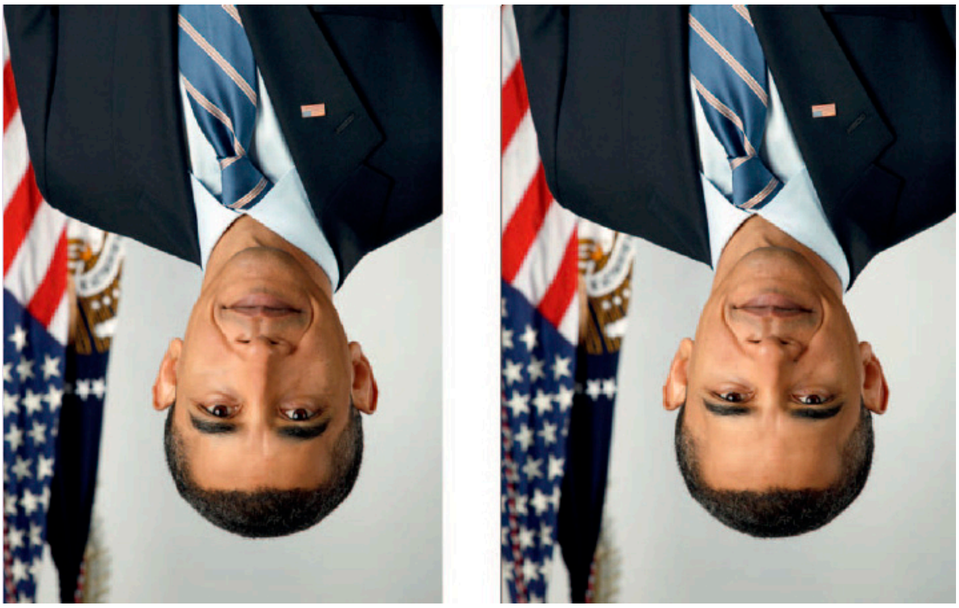

Figure 1. Holistic representation demonstrated by the face inversion effect. Look at the pictures; what you perceive is a holistic representation recalled from your memory. Turn the page upside down and look again.

The computational model outlined in Section 2.2.1 is a simplified version of the biological model we just described. To keep it simple, we focus on handwritten digit recognition instead of face recognition. While this recognition task does not have some of the challenges of a face recognition task, such as detection, illumination, occlusion, aging, expression changes, and so on, it does have a few challenges, such as, orientation, in-class variations (Figure 2) to name a few. Thus, a sound engineering model is still necessary to achieve high prediction accuracy. To accurately replicate the biological model, our computational model must decompose the recognition task into two simpler tasks: The first sub-system is specialized in taking as input a digit in any configuration or orientations, and transforms it into a holistic representation with a default orientation learned using supervised learning. The second sub-system uses the holistic representation generated by the first network to perform the recognition. Once we have a model that can take any variation of handwritten digits and transform it into a standard template, classifying the resulting template becomes trivial and can be done with any weak classifier.

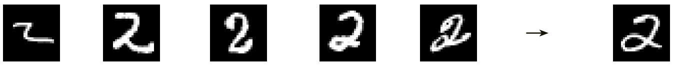

Figure 2. Holistic representation generated by the VAE. From a variety of images of the same class, shown on the left.

The remaining of the paper is organized as follows. Section 2.1, explores related work, Section 2.2 provides a detailed overview of the Variational Autoencoder (VAE) network architecture used to transform the input. In Section 3, we explain the dataset used in the experiment, the training and analysis of the experimental results. In Section 4, we discuss how to interpret the result. We conclude in Section 5.

2.1. Related Work

This section reviews related work in the area of invariant feature extractions, holistic processing and holistic representation of images. The idea of performing recognition using a holistic approach is not new. The oldest holistic model is the Principle Component Analysis (PCA) first proposed by Pearson in 1907 [16]. It a statistical-based approach that creates an average or Eigenface based on the entire training images express in a reduced format or dimension. Recognition is performed by representing the sample input as a linear combination of basis vectors and comparing it against the model. PCA operates holistically without any regard to the "specific feature of the face." PCA performance suffers under the non-idea condition such as illumination, different background, and affine transformations [17]. Other techniques seek to rectify this shortcoming by trying to integrate feature extraction in a manner that is explicit and, more importantly, invariant to affine transformation. Popular methods are morphological operations (i.e., thresholding, opening, and closing), pre-engineered filters (i.e., Scale-Invariant Feature Transform (SIFT) and Haar-like features), dynamically calculated filters using Hu transforms and learned filters using a Convolutional Neural Network [18].

The authors of [19] present a handwritten digit classifier that extracts features (i.e., contours) using morphological operations, then performs classification by computing the similarity distance of the Fourier coefficients from those features. The recognition rate for 5000 samples, provided by ETRI (Electronic and Telecommunications Research Institute), was 99.04%.

In [20], feature extraction that is invariant was achieved by using filters that have invariant properties as well as by "computing features over concentric areas of decreasing size." The authors used both normalized central moments and moment invariant algorithms to compute invariant feature extractors as a pre-processing step to object recognition. As many as 12 normalized central moments are needed to calculate filters invariant to scale; however, these filters are not invariant to rotate. The authors also used another 11 independent moments, which are based on Flusser's set and Hu's moment invariant theory. These filters are scale and rotational invariant. Once the features were extracted, the classification was done using a modified version of the AdaBoost algorithm. Misclassifications were reported as high as 56.1% and as low as 9.5% [20]. The disadvantage of engineer filters is their inability to generalize.

Another method of extracting features is where the feature extractor learns the appropriate filters. The model trainable feature extractor in [21] was based on the LeNet5 convolutional network architecture. To improve the recognition performance, the authors used affine transformations and elastic distortions to create additional tanning data, in addition to the original 60,000 training data from the Modified National Institute of Standards and Technology (MNIST) dataset [22]. The result of the experiments reported a recognition rate of 99.46%.

The architecture most similar to our model, is the learning model proposed by [23]. The architecture is an encoder-decoder model, which encode the feature using four binary numbers (the code), and a decoder that reconstructs the input based on that code. The objective was to create a model that can learn invariant features in an unsupervised fashion. The authors stated that the invariance comes from max-pooling, a method that was first introduced in the CNN [18] and by learning a feature vector (i.e., the code), using the encoder-decoder architecture, that incorporates both "what" is in the image and "where" it is located. The model achieved an error rate of 0.64% when trained using the full training set of the MNIST dataset.

To develop a computational model that mimics how our brain performs recognition, we use a generative model, follow by a classifier. While a Generative Adversarial Network (GAN) or a vanilla autoencoder would be suitable generators, a variational autoencoder is ideal because it matches the biological model's ability to learn invariant features. In this paper, we propose an architecture that is based on a variational autoencoder, capable of learning features that are invariant to all the transformations. Instead of reconstructing the original digit image in its original orientation, the model learns to generate a holistic representation of the image at a 0° position. By using a VAE instead of a vanilla autoencoder, the network learns a distribution of encoding to represent each digit; as a result, it generalizes better [23,24].

2.2. The Network Architecture

Figure 3 shows a high-level view of our proposed model. The model decomposes the recognition tasks into two sub-tasks: a generator for feature extractions and generation and a Neural Network for classification.

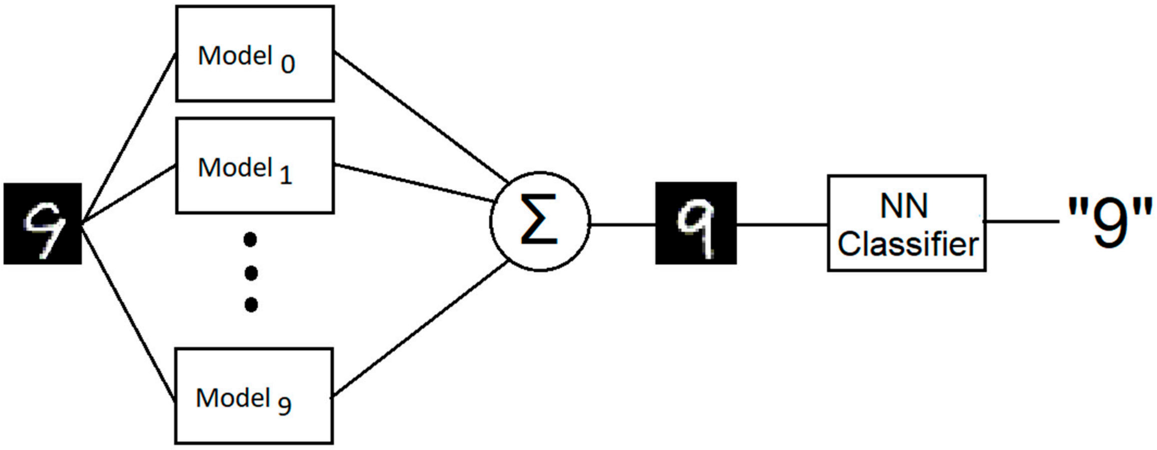

Figure 3. High-level view of the architecture. There are ten models/generators; each specializes in reconstructing a specific digit. The output of each generator is either the holistic representation of the digit it was trained for or the blank image. A composite image is constructed by summing the output of all models. The composite image is fed to a neural network for classification.

2.2.1. The Generator

The generator is based on the variational autoencoder architecture. The raw input image is 28 × 28 pixels, it is normalized and fed to the first convolution layer, which has 16 feature maps, 7 × 7 kernels and followed by a Rectified Linear Unit (RELU) activation function. Convolutions between the input image and the kernels are performed. Padding is used to retain the size of the input image. This step is followed by a non-linearity and a 2 × 2 max pooling, which produces a 14 × 14 image.

The output of the first convolution layer is fed to the second convolution layer with 32 feature maps, 7 × 7 kernels. Convolutions are performed again with padding so that the size of the input image is retained, followed by an RELU non-linearity and 2 × 2 max pooling. The resulting output is a 7 × 7 image.

The output of the convolutional layers is flattened and fed to a fully connected encoder neural network. The encoder is made of two layers of 512 and 128 neurons, respectively, with a Tanh activation function. The output of the encoder is fed in parallel to two layers made of 7 neurons each, referred to as mean and variance. These layers are used to calculate the compress representation of the input image, also referred to as the probabilistic latent vector Z, or "the code." This layer is also a vector of size 7, with the goal of representing the extracted features as a probability distribution. It is calculated as follows:

$$Z = \mu + \varepsilon \odot \mathcal{N}(0,\ 1) \tag{1}$$

where μ is the mean, ε is the variance, $\odot$ is an element wise multiplication, and $\mathcal{N}(0,\ 1)$ is a zero mean and unit variance gaussian distribution.

The chosen distribution is known to be tractable, and the latent vector Z will be forced to follow that distribution through K-L divergent. Our objective for using variational inference, as outlined in the next section, is strictly to learn the distribution of the latent vector Z, not for classification, visualization or data augmentation. The chosen distribution is held constant throughout testing. For more information on VAE, see [24].

The latent vector is fed to a decoder made of two layers with 128 and 512 neurons, respectively, with an RELU activation function. The final layer is the output layer of 784 neurons with a sigmoid activation function. The generator is shown in Figure 4.

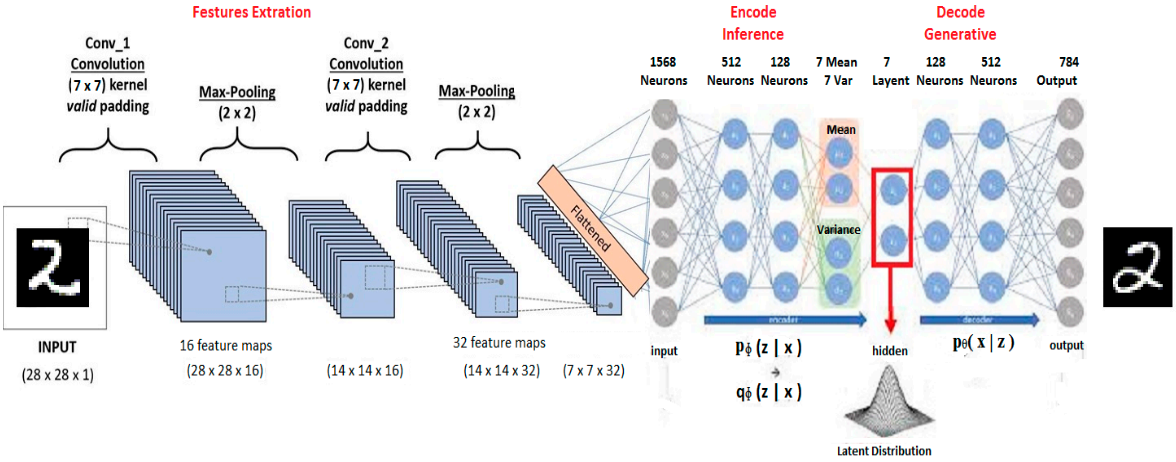

Figure 4. Generator model, based on a variational autoencoder with 2 convolutional layers for feature extraction, an encoder, and a decoder network.

The hyperparameters for the model were chosen through experiments and are determined to be optimal. For instance, we found that using 2 convolution layers with 8 kernels in the first layer and 16 kernels in the second layer was insufficient to produce good results, while using 32 kernels in the first CNN and 64 in the second produced diminishing returns. Experiments with a latent vector of size 2 or 3 impeded the network ability to converge. On the other hand, any latent vector size above 8 was more than needed.

2.2.2. The Classifier

The output vector of each generator is summed and fed to a classifier. The classifier is a neural network with one hidden layer of 256 neurons with a Tanh activation function and an output layer with ten neurons, one for each of digit class. A SoftMax activation is used in the output layer. The Softmax function express the output of the classifier, as a level of confidence that the given input belongs to that class. The output with the highest probability represents the network prediction. The sum probability of all the output is equal to one. The Softmax function is given by:

$$\overleftarrow{\mathrm{H}}_{\mathrm{jk}}(\mathbf{x}) = \frac{\mathrm{e}^{\mathrm{H}_{\mathrm{jk}}(\mathbf{x})}}{\sum_{\mathrm{l}=1}^{\mathrm{N}}\left(\mathrm{e}^{\mathrm{H}_{\mathrm{jl}}(\mathbf{x})}\right)} \tag{2}$$

The objective was to use a weak final classifier so that the overall performance of the architecture is attributed solely to the model. The parameters used in the final classifier NN were chosen because they are standard for a weak classifier and the performance of such a network is well established.

2.2.3. Variational Autoencoder

Variational Autoencoder architecture is based on autoencoders. It has an encoder and decoder just like autoencoders; however, VAE encodes each feature using a distribution, which allows the network to generalize better.

So, the goal of the variational autoencoder is to find a distribution $\mathrm{P}_{\theta}(\mathbf{z}|\mathbf{x})$ of some latent variables, so that new data can be sampled from it.

Using Bayesian statistics, we have:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)} \tag{3}$$

$$p(x) = \int_z p(x|z)p(z)dz \tag{4}$$

Given that the latent space, z, can be any dimension, we would have to calculate the integral for each dimension. Therefore calculating **p**(**x**), the marginal likelihood, is not tractable.

If we cannot compute **p**(**x**),then we cannot compute **p**(**z|x**).

To solve this problem, we use variational inference to approximate this distribution. To find an approximation to the truly intractable $p_\theta(z|x)$, we will introduce a known (tractable) distribution $q_\phi(z|x)$ and force $p_\theta(z|x)$ to follow this distribution, without interfering with the generation of the holistic image. We will use a method known as K–L divergent [24]. K–L divergent is a method for measuring similar distributions. The goal is to find ϕ that makes q closest to p_θ (see Figure 5).

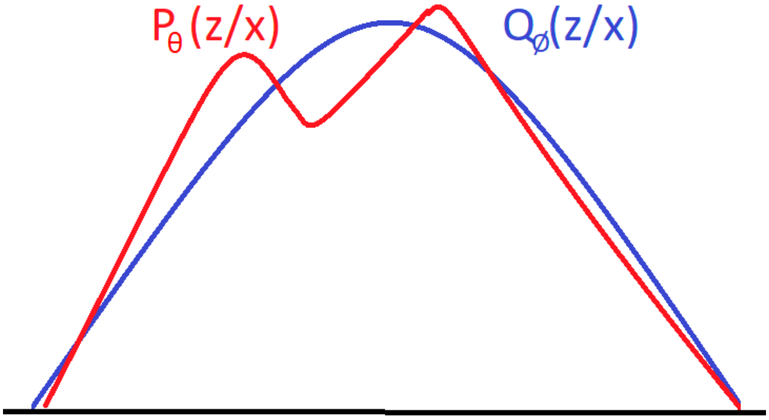

Figure 5. Approximating *p* given *q*.

The formula for K–L divergent is:

$$D_{kl}[\, q(x) \| p(x)] = \sum q(x) \log\left(\frac{q(x)}{p(x)}\right) \tag{5}$$

If we replaced *p(x)* and *q(x)* with conditional probabilities, we got:

$$D_{kl}\Big[Q_\phi(z|x) \| P_\theta(z|x)\Big] = \sum q(z|x) \log\left(\frac{q(z|x)}{p(z|x)}\right) \tag{6}$$

The equation is rewritten as an expectation:

$$D_{kl}\Big[Q_\phi(z|x) \| P_\theta(z|x)\Big] = E_{z\sim Q_\phi(x|z)}[\, \log\left(\frac{Q_\phi(z|x)}{P_\theta(z|x)}\right)] \tag{7}$$

Substituting $P_\theta(z|x)$ with Equation (3) using Bayes' and log rules, we got:

$$\begin{aligned} &D_{kl}[\, Q_\phi(z|x) \| P_\theta(z|x) \\ &= \log p(x) - E_z[\log(P_\theta(x|z))] + D_{kl}[Q_\phi(z|x) | | P_\theta(z)\Big] \end{aligned} \tag{8}$$

Since D_{kl} is always positive, then we can conclude,

$$\text{Log } P(x) \; >= \; E_z[\log(P_\theta(x|z))] - D_{kl}[Q_\phi(z|x) | | P_\theta(z)\Big] \tag{9}$$

Therefore, if we maximize the term on the right-hand side, we are also maximizing the term on the left-hand side. This is why the term on the right-hand side is called estimate likelihood lower bound (ELBO). Likewise, if we minimize $D_{kl}[Q_{\phi}(z|x)||P_{\theta}(z)]$ (because of the minus sign), then we are maximizing $E_z[log(P_{\theta}(x|z))]$ where E is the Expectation of a certain event occurring.

Therefore, the loss function for variational autoencoder is:

$$L(\theta,\ \phi) = -E_{z\sim Q_{\phi}(x|z)}\ [\log(P_{\theta}(x\Big|z)]\ +\ D_{kl}\left(Q_{\phi}(z|x)\Big\|P_{\theta}(z)\right) \tag{10}$$

So, to compute $D_{kl}[Q_{\phi}(z|x)||P_{\theta}(z)]$, we estimate the unknown distribution with the known distribution $\mathcal{N}(0,\ 1)$.

Suppose we have two multivariate normal distributions defined as:

$$P(x) = \mathcal{N}(x,\ \mu_1,\ \varepsilon_1) \tag{11}$$

$$Q(x) = \mathcal{N}(x,\ \mu_2,\ \varepsilon_2) \tag{12}$$

where μ_1 and μ_2 are means, and ε_1 and ε_2 are the covariance matrix or variance.

The multivariate normal density distribution of dimension k is defined as:

$$\mathcal{N}(x,\ u,\ \varepsilon) = \frac{1}{\sqrt{(2\pi)^k|\varepsilon|}}e^{-0.5\ (x-\mu)^T\varepsilon^{-1}(x-\mu)} \tag{13}$$

$$\begin{aligned} &D_{kl}(p(x)\|q(x)) \\ &= \left[\ \log\tfrac{|\varepsilon_2|}{|\varepsilon_1|} - d\ +\ \text{trace}\left(\varepsilon_2{}^{-1}\ \varepsilon_1{}^{-1}\right)\right. \\ &\left.+\left(\mu_2 -\ \mu_1\right)\varepsilon_2{}^{-1}\left(\mu_2 -\ \mu_1\right)\right] \end{aligned} \tag{14}$$

If we set one of the distributions to be zero mean and the unit variance as $\mathcal{N}(0, 1)$, then:

$$D_{kl}[\mathcal{N}(\mu,\ \varepsilon)\ \|\ \mathcal{N}(0,\ 1)] = \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\varepsilon^2\right) -\ \mu^2 -\ \varepsilon^2\right) \tag{15}$$

To solve the term $E_{z\sim Q_{\phi}(x|z)}\ [\log(P_{\theta}(x|z)]$ in Equation (10), we use re-parameterization trick.

Re-Parameterization

To minimize a loss function, we take the partial derivative with respect to one of the model parameters and set it to zero. It turns out that it is difficult to take the derivative of this loss function with respect to ϕ, because the expectation is taken over the distribution, which is dependent on ϕ. This is why a re-parameterization trick is required. Re-parameterization involves using a transformation to rewrite the expectation $E_{z\sim Q_{\phi}(x|z)}$ in such a way that the distribution is independent of the parameter ϕ. So, an expectation of the form $E_{z\sim Q_{\phi}(x|z)}\ [f(z)]$ can be re-written as $E_{P()}\left[f\left(G_{\phi}(\epsilon,\ x)\right)\right]$. $P(\epsilon)$ (reads probability of epsilon), is sampled from a gaussian distribution with zero mean and unit variance (i.e., $\epsilon \sim\mathcal{N}(0,\ 1)$). If we set $G_{\phi}(\epsilon,\ x)$ to be equal to a standard linear transformation $\mu_{\phi}(x) +\ \odot\ \sqrt{\varepsilon_{\phi}}$, then we can obtain a gaussian function of arbitrary mean and variance, $\mathcal{N}(\mu(x),\ \varepsilon(x))$, from the gaussian $\mathcal{N}(0,\ 1)$. This means we are transforming $\mathcal{N}(0,\ 1)$ to $\mathcal{N}(\mu(x),\ \varepsilon(x))$. Therefore, instead of sampling z from $Q_{\phi}(x|z)$, we instead sampled it from $G_{\phi}(\epsilon,\ x)$.

$$E_{z\sim Q_{\phi}(x|z)} =\ E_{P(\varepsilon)}\left[f\left(G_{\phi}(\varepsilon,\ x)\right)\right] \tag{16}$$

$$\begin{gathered} E_{z\sim Q_{\phi}(x|z)}\ [\log(P_{\theta}(x|z)] \equiv E_{Z\ \sim\ P(\varepsilon)}\left[\log\ P_{\theta}\left(x\ |z^l\right)\right] \equiv \\ \tfrac{1}{L}\sum_{l=1}^{L}\log\ P_{\theta}(x\ |z^l) \end{gathered} \tag{17}$$

where z^l comes from the transformation $\mu_\phi(x) + * \sqrt{\varepsilon_\phi}$ and is sampled from $\mathcal{N}(0,\ 1)$. This allows us to obtain the Monte-Carlo estimate of the expectation in a form that is differentiable. For more information on re-parameterization, we refer the reader to [24].

The K-L Divergent Loss Function

Plugging (15) and (17) into (10), give the complete VAE loss function:

$$\begin{aligned} L(\theta,\ \phi) = -\tfrac{1}{L}\sum_{l=1}^{L} \log P_\theta(x\,|z^l) \\ + \tfrac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\varepsilon^2\right) - \mu^2 - \varepsilon^2\right) \end{aligned} \tag{18}$$

The first term is the data fidelity loss, and the second term is the K–L divergent [24].

The Generator Loss-Function

As stated earlier, there are ten identical generators, and each generator network is trained to generate a different holistic digit. To prevent the network from learning the identity function, each network is trained with both similar digit and dissimilar digits. In every batch, half of the training data are composed of the digit the network is learning to generate, and we will refer to them as a similar digit. The other half of the training data is randomly chosen from the full training dataset (excluding similar digit), we will refer to them as dissimilar digits. As a result, we replace the data fidelity term in Equation (11), with the contrastive loss function [25]. With the usage of a conditional parameter Y, the contrastive loss function accommodates two optimization paths, one when the ground truth is similar to the input and another when they are dissimilar. For more information on contrastive loss, we refer the reader to [25].

Let x and x represent the ground truth (see Figure 6) and the generated images, respectively.

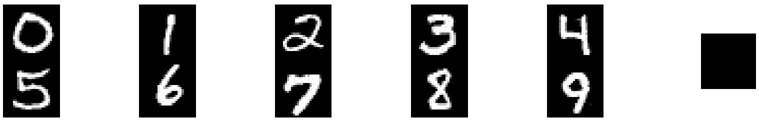

Figure 6. Ground truth labels are used to train the generator. For example, the network being trained to recognize or generate the digit "1" is given the ground truth image for "1" and the blank image for all other digits.

Let Y be a binary label assigned to this pair, such that Y = 1 when x and x̃ are expected to be similar, and Y = 0 otherwise.

The absolute distance (or L1Norm) is given by:

$$D_\theta = \sum_{l=1}^{L} \left| x - x \right| \tag{19}$$

$$L(\theta) = \frac{1}{2} Y\, D_\theta{}^2 + \frac{1}{2}(1 - Y)\ (\mathrm{Max}[0,\ m - D_\theta])^2 \tag{20}$$

The Contrastive loss as defined in [25] is:

The margin m, defines the radius around which dissimilar digits contribute to the loss function. When Y = 0 and m < D$_\theta$, then no loss is incurred. Since the objective is for dissimilar digits to contribute equally to the learning process, m is set to a number greater than one (i.e., m = 5).

Replacing the reconstruction likelihood $\frac{1}{L}\sum_{l=1}^{L}\log P_{\theta}(x\,|z^{l})$ in Equation (18) yields:

$$L\,(\theta,\ \phi) = -\frac{1}{2}Y\,{D_{\theta}}^{2} + \frac{1}{2}(1-Y)\,(\mathrm{Max}[0,\ m-D_{\theta}])^{2} + \frac{1}{2}\sum_{j=1}^{J}\left(1+\log\left(\varepsilon^{2}\right)-\ \mu^{2}-\ \varepsilon^{2}\right) \tag{21}$$

3. Dataset and Training Strategy

The results of the experiments using the MNIST database [22] are outlined in this section. The database is a well-known benchmark with 60,000 training images and 10,000 test images. The images are 28 × 28 pixels gray-level bitmaps with digits centered in a frame in white with a black background. As a result, digit detection is not necessary, since the digit is the only object in the image. The entire 60,000 images are used as training inputs to train the generator. The ground truth images that are used as labels are carefully chosen from the training set to be as differentiable as possible (Figure 6). They are the holistic representation the generator is expected to reconstruct. Another set of 50 images, five from each class, were randomly chosen from the training set, and are used as a validation set.

To reduce the resource necessary to train the network, the model uses a separate network for each digit without sharing model parameters. Each generator network is trained separately for 200 epochs using the full 60,000 training set. Due to the probabilistic nature of the network, some digits required more training than others. Starting at epoch 100, the model's progress is monitored using the validation set, and a snapshot of the model is saved for all 100% success rate. Success in this context is defined as the model's ability to reconstruct the digit being trained for or produce a blank image for all other digits (see Figure 3). Once training is over, a stable checkpoint is chosen for further integrated training. A model is deemed stable if, during training, it has at least 3 consecutives success using the validation set.

The classifier Neural Network is trained separately for 50 epochs on the full training dataset.

4. Results and Discussion

We calculate the success rate S (or accuracy) as the total number of digits correctly predicted by the network or true positive (TP), divided by the number of total numbers of test data (TD). Therefore:

$$\mathbf{S} = \frac{\mathbf{TP}+\mathbf{TN}}{\mathbf{TD}} \tag{22}$$

Consequently, the error rate ER is given by (1—S) or:

$$\mathbf{ER} = \frac{\mathbf{FP}+\mathbf{FN}}{\mathbf{TD}} \tag{23}$$

The integrated architecture achieves a 99.05% accuracy in the recognition task, which is near state of the art, when compared to the best-published results on this dataset [26] and done so without augmenting the dataset or pre-processing. Since the integrated architecture summed the output of all the individual models, any false positive error resulted in an overlap representation of two or more holistic images. As a result, the composite image is un-recognizable by the final classifier (see Figure 3). This adversely affects the accuracy of the integrated architecture. To demonstrate the model's superior ability to recognize and reconstruct the digit for which it is trained for, ten more experiments were conducted, one for each digit. The results show an overall true positive accuracy of 99.5%. Table 1 shows the raw data of the experiments. For instance, the test conducted on the model that was trained for digit seven yields an accuracy of 99.84%. As shown in Table 1, the test set contains a total of 1028 images for digit seven (similar digits or expected true positive) and 8972 images for the other digits (dissimilar digits or expected true negative), 10,000 test images all together. The result shows 6 cases when the model erroneously generated a blank image for a valid input image of the digit seven (false negative) and 10 cases when it generated the holistic representation of digit seven, when in fact the input image was not the digit seven (false positive). Given this information and Equation (22), the success rate for the model trained for digit seven is computed as follows:

$$S = \frac{1018 + 8966}{10000} = 99.84\%.$$

Table 1. Individual model experimental results.

	Model 0	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**Model 8**	**Model 9**	**Overall**
Similar Digits	980	1135	1032	1010	982	892	958	1028	974	1009	10,000
Dissimilar Digits	9020	8865	8968	8990	9018	9108	9042	8972	9026	8991	90,000
False Negative	0	0	0	7	9	6	7	6	5	10	0.50%
False Positive	6	8	16	8	7	12	10	10	13	13	0.11%

The lowest accuracy recorded was 99.77% for the model trained for digit nine and the best accuracy was 99.94% for the model trained for digit zero (or error rates from 0.23% to 0.06%).

The most extensive and recent survey of recognition models using the MNIST and the Extended MNIST (EMNIST) dataset was published in the paper, "A Survey of Handwritten Character Recognition with MNIST and EMNIST". Among them are 44 models that achieve accuracy similar to ours, that is accuracy greater than 99% (or error < 1%) using the MNIST dataset without data augmentation or preprocessing. The error rates reported ranged from 0.95% to 0.24% [26]. Among them, 37 use CNN for feature extraction followed by an algorithm and or a neural network to perform the classification. The fact there are no simpler model (i.e., less than two convolution layers and fewer kernels) with results similar to ours confirms our hypothesis.

The handwritten digit recognition was chosen to demonstrate the model because it is a much simpler problem than face recognition. The logical next step is to apply it to face recognition.

5. Conclusions

This paper proposed a classifier for handwriting digit recognition based on a model that first transforms the input to a holistic or average representation and then performs classification based on that template. The experimental results using the MNIST dataset show that such a strategy allows the network to learn features that are invariant to all the transformations included in the dataset. The integrated architecture achieved near state-of-the-art recognition, and the model demonstrated a superior ability individually to recognize and reconstruct digits, when compared to the best-published results on this database, and did so without pre-processing or augmenting the dataset.

Author Contributions: The work has been primarily conducted by M.J. under the supervision of K.E. Discussions about the algorithms and techniques presented in this paper took place among the authors over the past year. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors acknowledge the University of Bridgeport for providing the necessary resources to carry this research conducted under the supervision of Khaled Elleithy. The cost of publishing this paper was funded by the University of Bridgeport, CT, USA.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Civile, C.; McLaren, R.P.; McLaren, I.P. The face inversion effect—Parts and wholes: Individual features and their configuration. *Q. J. Exp. Psychol.* **2014**, *67*, 728–746. [CrossRef]
2. Yin, R.K. Looking at upside-down faces. *J. Exp. Psychol.* **1969**, *81*, 141. [CrossRef]
3. Yovel, G.; Kanwisher, N. Face perception: Domain specific, not process specific. *Neuron* **2004**, *44*, 889–898.
4. Carey, S.; Diamond, R. From piecemeal to configurational representation of faces. *Science* **1977**, *195*, 312–314. [CrossRef]

5. Passarotti, A.M.; Smith, J.; DeLano, M.; Huang, J. Developmental differences in the neural bases of the face inversion effect show progressive tuning of face-selective regions to the upright orientation. *Neuroimage* **2007**, *34*, 1708–1722. [CrossRef] [PubMed]
6. Searcy, J.H.; Bartlett, J.C. Inversion and processing of component and spatial–relational information in faces. *J. Exp. Psychol. Hum. Percept. Perform.* **1996**, *22*, 904. [CrossRef]
7. Fodor, J.A. *The Modularity of Mind*; MIT Press: Cambridge, MA, USA, 1983.
8. Farah, M.J.; Wilson, K.D.; Drain, H.M.; Tanaka, J.R. The inverted face inversion effect in prosopagnosia: Evidence for mandatory, face-specific perceptual mechanisms. *Vis. Res.* **1995**, *35*, 2089–2093. [CrossRef]
9. Leder, H.; Bruce, V. When inverted faces are recognized: The role of configural information in face recognition. *Q. J. Exp. Psychol. Sect. A* **2000**, *53*, 513–536. [CrossRef]
10. Taubert, J.; Apthorp, D.; Aagten-Murphy, D.; Alais, D. The role of holistic processing in face perception: Evidence from the face inversion effect. *Vis. Res.* **2011**, *51*, 1273–1278. [CrossRef]
11. DeGutis, J.; Wilmer, J.; Mercado, R.J.; Cohan, S. Using regression to measure holistic face processing reveals a strong link with face recognition ability. *Cognition* **2013**, *126*, 87–100. [CrossRef]
12. Rossion, B. Distinguishing the cause and consequence of face inversion: The perceptual field hypothesis. *Acta Psychol.* **2009**, *132*, 300–312. [CrossRef]
13. Tanaka, J.W.; Farah, M.J. Parts and wholes in face recognition. *Q. J. Exp. Psychol.* **1993**, *46*, 225–245. [CrossRef]
14. Schwaninger, A.; Lobmaier, J.S.; Collishaw, S.M. Role of Featural and Configural Information in Familiar and Unfamiliar Face Recognition. In Proceedings of the International Workshop on Biologically Motivated Computer Vision, Tübingen, Germany, 22–24 November 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 643–650.
15. Grand, R.L.; Mondloch, C.J.; Maurer, D.; Brent, H.P. Impairment in holistic face processing following early visual deprivation. *Psychol. Sci.* **2004**, *15*, 762–768. [CrossRef] [PubMed]
16. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
17. Archana, T.; Venugopal, T. Face recognition: A template based approach. In Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Noida, India, 8–10 October 2015; pp. 966–969.
18. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
19. Jeong, C.-S.; Jeong, D.-S. Hand-written digit recognition using Fourier descriptors and contour information. In Proceedings of the IEEE Region 10 Conference. TENCON 99.'Multimedia Technology for Asia-Pacific Information Infrastructure'(Cat. No. 99CH37030), Cheju Island, Korea, 15–17 September 1999; pp. 1283–1286.
20. Barczak, A.; Johnson, M.J.; Messom, C.H. Revisiting moment invariants: Rapid feature extraction and classification for handwritten digits. In Proceedings of the International Conference on Image and Vision Computing New Zealand (IVCNZ), Hamilton, ON, Canada, 5–7 December 2007.
21. Lauer, F.; Suen, C.Y.; Bloch, G. A trainable feature extractor for handwritten digit recognition. *Pattern Recognit.* **2007**, *40*, 1816–1824. [CrossRef]
22. LeCun, Y.; Cortes, C.; Burges, C.J. The mnist database of handwritten digits. *Lecun. Com/Exdb/Mnist* **1998**, *10*, 34.
23. Ranzato, M.A.; Huang, F.J.; Boureau, Y.-L.; LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
24. Kingma, D.P.; Welling, M. Auto-encoding variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
25. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 1735–1742.
26. Baldominos, A.; Saez, Y.; Isasi, P. A survey of handwritten character recognition with mnist and emnist. *Appl. Sci.* **2019**, *9*, 3169. [CrossRef]