Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 3, pp. 549-558, August 2025

RESEARCH PAPER

OPEN ACCES

Few-shot Classification of Smartphone Photos using Hidden Markov Model and Siamese Network

Zulkarnaen Hatala[1], **Muhammad Hudzaly**[2]

[1] Department of Informatics, Politeknik Negeri Ambon, Ambon, Indonesia
[2] Faculty of Industrial Engineering, Institut Teknologi Kalimantan, Balikpapan, Indonesia

ABSTRACT

Images from the increasing use of smartphones are so large that they are nearly impossible to handle by hand. The problem arises when a person needs to classify these photos into groups or classes. Smartphones are low-performance devices in contrast to desktop or cloud-based computers. Many solutions of image classification using various types of Convolutional Neural Network (CNN) are performed on massive cloud-based supercomputers. These computers are often equipped with very high-end, additional specialized graphics processing units (GPUs) at remarkable prices. In fact, to implement classification in most smartphones currently on the market, we need an algorithm that requires less computation. Based on this fact, we propose an HMM that requires fewer parameters. This research aims to examine the HMM method for classification of photos taken with a smartphone. For comparison, we also outline the results from the Siamese CNN. The same data are used for training and testing for both models. For HMM, we use the Discrete Cosine Transform (DCT) to extract salient features of images. The number of training examples is very small compared to the test set. Here we carried out a few-shot classification method. In the training phase, we used the Maximum Likelihood (ML) criterion-based Baum-Welch algorithm. Two versions are used; isolated training is applied first and later followed by jointly-embedded Baum-Welch estimation of parameters. For recognition of the HMM, Viterbi algorithm is applied. Performances of both procedures were measured. Based on the test results, HMM achieves 0,94 precision, 0.85 recall, F1 score 0.89, and accuracy 0.90 while Siamese claims 0.87, 0.98, 0.92, and 0.91. The result shows that HMM, which has an advantage over Siamese in terms of fewer parameters, still competes with Siamese CNN with only slight decrease in performance. We conclude that HMMs are suitable over Siamese CNNs to be implemented in low-performance devices such as cellphones.

PAPER HISTORY
Received June 10, 2025
Revised August 21, 2025
Accepted August 28, 2025
Published August 30, 2025

KEYWORDS
Few-shot Photo Classification;
Hidden Markov Model;
Discrete Cosine Transform;
Siamese Network

AUTHOR EMAIL
dzulqarnaenhatala@gmail.com
m.hudzaly@lecturer.itk.ac.id

I. Introduction

Nowadays, photographic events are easily and freely captured using a mobile phone. A camera inside a smartphone is capable of shooting photos better and easier anywhere and anytime at no cost. Taking pictures has become a habit for people around the world. A single person can have an average of 640 photographs (photos) taken by his (or her) smartphone. Besides its function as a camera, a smartphone integrates its capability for further processing and organizing of document images. A smartphone user with the profession of an educator, such as a professor, lecturer, or teacher, can access digitally the student-submitted assignments using a smartphone. The advantage of using a phone over a dedicated flatbed paper scanner is mobility and flexibility. With an easy-to-use camera-based phone, resulting in so many images taken for each person, surely there is a need to efficiently manage and organize these images for many extended purposes. A professor needs to retrieve all students' papers and distinguish them from other types of photos, such as panoramas, family events, ceremonials, etc. Since the images in smartphones are massive, even if the classification of similar images into their category is quite easy, the repetition process could be exhaustive to perform by humans. Smartphones and computers on the other hand can be programmed to perform classification automatically. They are faster and do not suffer from tiredness. Furthermore, the person is free to move his or her images into a better processing device like a high-performance laptop to achieve lesser execution time and higher accuracy. The classification task of images falls into methods in the field of machine learning. These images taken by smartphones are rather challenging to recognize than the ones acquired using a dedicated flatbed scanner. This is because smartphone photos lack regularity and are distorted by the movement of human hands. The images also suffer from various lighting conditions.

Hidden Markov Model (HMM) [1], [2], [3] is a statistical method that has been used in many real-world applications. Historically, HMM has gained success in the field of speech-related topics, including recognition[2] and verification [4]. Lately, HMM has been applied in many images classification research. HMM applied to face recognition to achieve reasonable computation was elaborated in [5]. HMM, in conjunction with Neural

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

Network (NN), Principal Component Analysis (PCA), and Gabor Filter (GF) is explored in [6]. The classification of small, low-resolution thumbnail images is done in [7].

On the other side, Neural Networks are also used extensively in image classification. Classification with very small samples in comparison to test samples is done with many models from various architectures. Residual Networks (ResNet) are used in [8]. An adversarial network is used in [9]. Transformers are used in [10], [11] [12]. In [13] the authors use EfficientNet-B7 to segment and classify breast cancer from thermal images. Siamese twin network architecture is used to classify handwritten alphabets [14]. Notably the use of CNN involves many parameters of the model. The problem with CNN families is their large number of parameters that are hardly, or even impossible to implement in low-performance devices. Smartphones are low-performance devices in contrast to desktop or cloud-based computers. Many solutions with CNNs are performed on massive cloud-based supercomputers and often equipped with very high-end additional specialized graphics processing units (GPUs). Of course, these GPUs are mostly associated with remarkable prices that are not cheap to purchase. In fact, to implement classification in most smartphones currently on the market, we need an algorithm that has less computation The motivation to choose HMM instead of CNN is the computing power which the model requires. So, this study should show the adequate performance by HMM despite its smaller number of parameters.

In this paper, the images taken using smartphones are categorized using two different approaches mentioned above, the Hidden Markov Model and Neural Network. When using HMM, we use the Discrete Cosine Transform (DCT) [15] as a feature extraction technique. The small number of DCT coefficients should capture a simple salient feature that mimics the human recognition process or even further delicate patterns missed by human eyes. The local similarity of neighboring rows and columns shows dependencies of inter-neighboring blocks inside an image. These facts motivate the classification of photographs using HMM. Another approach we used is a Siamese CNN, a type of Neural network with Deep Learning [14] which is also applied in this research as a comparison to HMM.

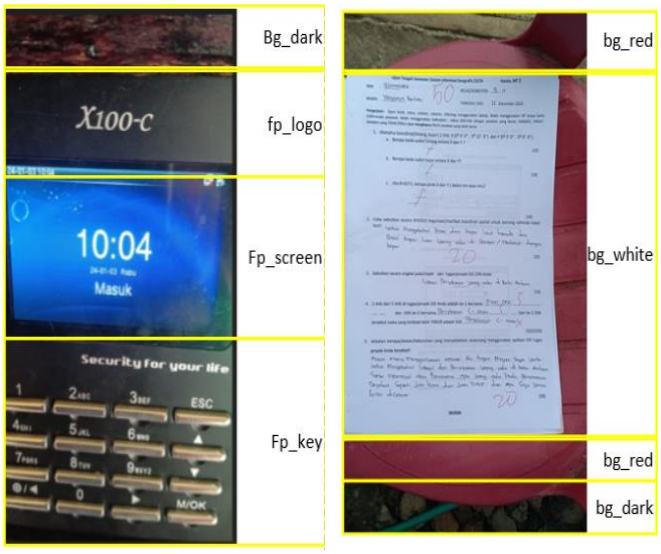
The main contribution of this paper is to describe how the HMM method makes it possible the implementation of photo classification process in low-performance device cellphones. In the next section II, we describe about dataset and two methods, HMM against Siamese. In Section III we provide the results from two methods. Next in section IV, we discuss the results, limitations and further research. Finally, we conclude the findings in section IV.

II. Materials and Methods

Below, we describe the data or samples used in this experiment. Also, we describe two methods, HMM and Siamese CNN, to perform feature extraction, training, and classification on the same training set and test set.

A. Dataset

Images taken using a smartphone camera are employed in this research. The smartphone used features a triple rear camera setup with a 50MP main sensor, a 2MP macro lens, and a 2MP depth sensor. Main Camera has 50MP, f/1.8 aperture, PDAF, Macro Lens has 2MP, f/2.4 aperture, Depth Sensor has 2MP, f/2.4 aperture. All images are taken using WhatsApp software, a social media application. The smartphone with this camera specification can be purchased at regular average prices. The owner of the phone is a lecturer by profession. We limited the kind of images to only two categories, fingerprint images in Fig. 1 (a) and student submission papers in Fig. 1 (b). An employee must log their finger into the biometric machine every workday at the university. Since the university has not yet published the log confirmation, this worker also takes a photo of taken fingerprint for their own proof. This can help prevent contradiction between employees and management staff. In the second case, an educator regularly runs paper-based tests for a class exam session. Whenever they complete marking the paper, they shoot a photo of the submission and secure it in their smartphone. The digital version of the paper can be sent back to the student. By sending back the paper to the students, the teacher guarantees transparency and fairness among all participants. The data collection examples of these images are shown in Fig. 1.



(a) (b)

Fig. 1. Two classes of smartphone photos (a) finger print machine, (b) Student answer sheet

We use a total of 114 samples, 56 of the photos belong to the student answer sheet, and the rest are of the board. A few images were chosen at random and labeled for training purposes. The separation between test and training data depends on the availability of the labels that belong to images. Only data with labels can be included in the training set. Currently, only 12 samples are labelled; each class has 6. The images that are labelled are listed in **Table** 1. The portions of sub sub-image that being labelled are depicted in Fig. 1.

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

Table 1. List of Training samples

No	Filename	Class
1	IMG-20240102-WA0000	finger machine
2	IMG-20240103-WA0000	finger machine
3	IMG-20240102-WA0002	finger machine
4	IMG-20240103-WA0002	finger machine
5	IMG-20240723-WA0001	finger machine
6	IMG-20240718-WA0005	finger machine
7	IMG-20240101-WA0047	answer sheet
8	IMG-20240101-WA0052	answer sheet
9	IMG-20240101-WA0113	answer sheet
10	IMG-20240101-WA0168	answer sheet
11	IMG-20240101-WA0152	answer sheet
12	IMG-20240101-WA0136	answer sheet

B. Data Preprocessing

Conversion of images into grayscale is performed before feature extraction by the HMM method. This is the only preprocessing applied in this research. Currently, none of the other techniques, such as noise reduction, contrast enhancement, or artifact removal, is involved. Besides rescaling to 280 pixels wide, the image is passed as is into the feature extraction stage in HMM or the input layers in Siamese. The choice of 280 pixels wide is intuitive. In this case, it is still big enough for humans to supervise. And the salient features are still rich. But also, the image is small enough to be processed quickly by cellphones.

C. Hardware, software, and tools

In this research, the HMM method is implemented on a computer with i3-7100U processor, 8GB RAM. The library for HMM is the Hidden Markov Toolkit (HTK) compiled for the Cygwin environment. While the hardware we used for the Siamese CNN is deployed on an i5-3470 8GB RAM plus GPU GTX 1050 TI 4GB. Libraries for Siamese is implemented using Python, TensorFlow, Keras, and CUDA with GPU support.

D. HMM Method

The very first step of data processing is image size reduction: photos are resized (reduced) proportionally to a fixed width and variable length. Most of the images taken by smartphones are much bigger than the reduced size. The scaling of image size is intended to achieve faster processing time and to fit the model. In this research, all images are resized to a constant width of 280 pixels. The heights are relaxed to vary depending on their original values to keep proportional rescaling. Then all images are converted into 8 8-bit gray-level image. Followed by feature extraction: an image is divided into subblocks with a constant block size. As shown in Fig 2, subblocks are shown that are not overlapped. Overlapping blocks are not yet used. A feature is extracted for every block and is assumed to be outputted by a state of an HMM. Discrete Cosine Transform (DCT) coefficients are calculated for each block [16], [17], [18]. The number of blocks for an image varies depending on the image length. As shown in Fig. 2., along the horizontal direction from left to right, we have C number of blocks for a row. On the vertical direction, we have total R blocks for a column from b_{00} until $b_{R\text{-}1C\text{-}1}$. So, the total number of DCT features for an image is RxC. In addition, at each end of a row, a special feature is added called a marker [19]This is positioned at imaginary block M in Fig. 2.

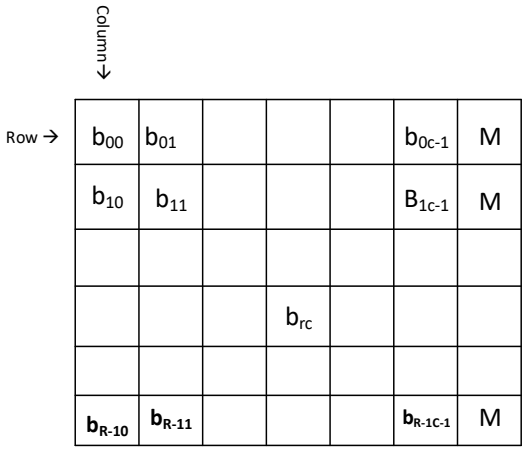


Fig. 2. Block based feature extraction

In this research, a block size 4x4 pixels is chosen with no overlap with other blocks. Suppose the subblock is a 4x4 matrix with element at row *i*, column *j* is *Gr(i,j),* then its DCT is also 4x4 in dimension as shown in Fig. 3. $F(u,v)$ is the element of DCT at Fig. 3 and is calculated as [16] in equation Eq. (1):

$$F(u,v) = \frac{C(u)C(v)}{2} \sum_0^3 \sum_0^3 \left[Gr(i,j) \cos\left(\frac{(2i+1)u\pi}{8}\right) \cos\left(\frac{(2j+1)v\pi}{8}\right)\right] \quad (1)$$

with

$$i,j,u,v \in \{0,1,2,3\},$$

u, v are coordinates of a DCT value, i, j are pixel coordinates of the sub-image, and $Gr(i,j)$ is a gray level value for that pixel, π is a ratio between the circumference of a circle to its diameter, *C(u)* and *C(v)* are scaling factors or magnitudes where their values are determined by *C(m)* below:

$$C(m) = \begin{cases} \frac{\sqrt{2}}{2} \text{ if } m = 0 \\ 1 \text{ otherwise} \end{cases}$$

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

As mentioned above, for any block b=b_{rc} in Fig. 2, can be calculated 16 DCT coefficients, of which only 6 are taken as features for block b as in equation Eq. (2) [20]:

$$\begin{aligned} f_{b0} &= F(0,0)\,, f_{b1} = F(0,1), \\ f_{b2} &= F(1,0), \\ f_{b3} &= \tfrac{1}{4}\textstyle\sum_{i=2}^{3}\sum_{j=0}^{1}|F(i,j)|\,, \\ f_{b4} &= \tfrac{1}{4}\textstyle\sum_{i=2}^{3}\sum_{j=0}^{1}|F(j,i)|\,, \\ f_{b5} &= \tfrac{1}{4}\textstyle\sum_{i=2}^{3}\sum_{j=2}^{3}|F(j,i)| \end{aligned} \tag{2}$$

In Eq. (2), f_{bi} is the i-th feature corresponding to subblock b and $F(u,v)$ as defined in Eq. (1). The frequency map of these coefficients is given in Fig. 3: When features have been extracted, then we train the HMMs: The topology of the HMM is often depicted as how many states and the transition between of its underlying Markov Chain[21]. In this study, HMMs are on the same topology (constant states number). These 5 states, left to right HMM are shown in Fig 4.

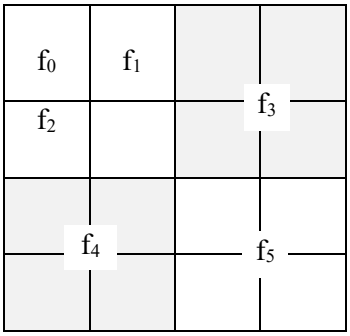


Fig 3. Frequency map of DCT features

Features along a row (horizontal direction of image) are assumed to be generated by HMMs in conjunction. Each HMM consists of s0 a start state, s1, s2, s3, the excitation states, and s4, an absorption state. The emission probability of a single feature vector

$$\boldsymbol{f} = (f_{b0}, f_{b1}, f_{b2}, f_{b3}, f_{b4}, f_{b5})$$

Conditioned on state s_j, $p(\boldsymbol{f}|s_j)$ There are six independent Gaussians. This probability is given by [22], [23] equation Eq. (3) :

$$p(\boldsymbol{f}|s_j) = \frac{1}{8\pi^3 \prod_{i=0}^{5}\sigma_i}\, e^{-3\sum_{i=0}^{5}\left(\frac{f_{bi}-\mu_i}{\sigma_i}\right)^2} \tag{3}$$

In equation Eq. (3), f_{bi} is the random variable associated with the i-th feature value as defined in Eq. (2), σ_i and μ_i are its standard deviation and mean, and e is Euler's number. These HMMs are to model all vectors of features along the rows and columns.

Firstly, a single initial HMM is initialized using the Viterbi training algorithm [24], [25], [26] employed over all training data. Even this step does not need a label for every photo, and the initial HMM can be initialized using all available test set, but in this research, we did not do it. This is because in real-world applications, the test data is assumed not yet exist. Initialization is performed using the HTK program **HInit**. Then this HMM is copied into five to six HMMS in sequence for each row. The number of HMMs along the row depends on the width of the image. Later, **HRest,** which perform the isolated Baum-Welch algorithm are used to train individual HMM based on the data. Isolated training is when Baum-Welch works on very tight assumptions of each HMM boundary. Based on the single label, it is assigned. As can be seen later, this is different from the **HERest** training, where the Baum-Welch altered the label boundaries by considering all data available. Finally, all HMMs are joined (embedded) into a single large HMM and trained using all training set available. Embedded training is performed using the HTK tool **HERest**. Based on HMM topology and emission probability, we counted a total of 49 smaller HMMs from two classes. These HMMs are jointly trained in this system. All rows in the single labelled region i.e "fp_log" in Fig.1 (a) share the same sequence of HMMs.

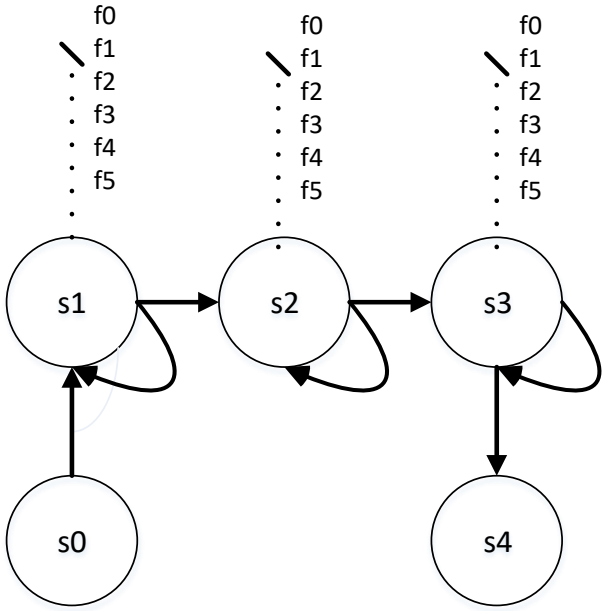


Fig 4. HMM Topology

When the final model is complete, we perform classification: the Viterbi algorithm, which is constrained to a recognition path for each category, is performed against test images. This variant of the Viterbi algorithm is called token passing [27], [28] . In case there are multiple recognition paths, the Viterbi recognition algorithm will determine the probability of each image against all possible paths. The test image is decided to be of a certain category if and only if it has the best Viterbi score to an associated path of that category. The HTK program **HVite** is used to perform a classification algorithm. **HVite** uses a grammar and a dictionary to perform its task. HMM uses separated feature extraction outside its training dan testing process, while CNN feature extraction is embedded in the training process.

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

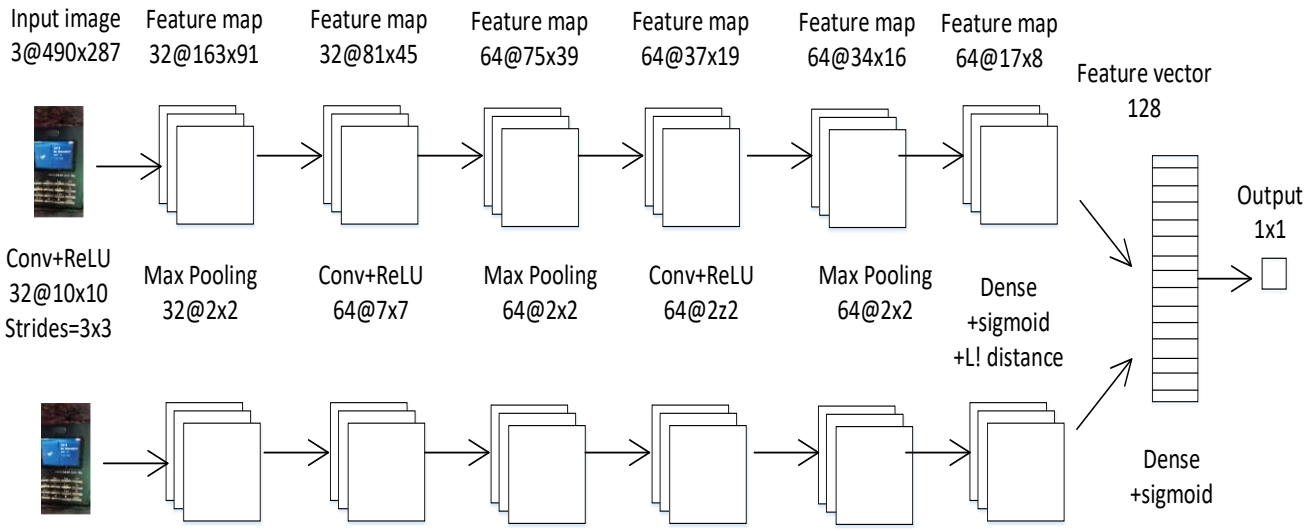


Fig. 5. Siamese architecture for photo classification

E. Siamese Neural Network method

To compare results against HMM, we used a Siamese network [14], [29] as shown in Fig. 5. The twin network is a type of CNN that applies deep learning [30] technique. The use of the multiple hidden layers approach, which was applied here, is mainly described in [14]. The method in [8] requires fixed dimensions of width and height of the photos. After resizing proportionally of the original images, the images are optionally cropped or padded on the left and right sides. In this way, the dimension of images forwarded into input layers is kept constant at 498x280 pixels. The color of the image is also passed as a 3rd dimension of the tensor [31]. This color information contains three channels for red, green, and blue (RGB) respectively. As images are passed to the input layer, they are subsequently processed through various layers. The Siamese network contains double parallel convolutional layers that perform automated feature extraction [32].

In Fig. 5, the input to the architecture is 2 queues contain each containing a sequence of photos that are sent to the first layer. This first layer is a convolutional layer with a kernel size = 10x10, several filters of 32 and strides = 3x3, with a ReLU activation function (32@10x10 text in the middle of the image). This result produces 32 feature maps with a size of 163x91 (32@163x91 is seen in the text position at the top of the image). After that, the output of the convolution layer is forwarded to the max pooling layer, producing 32 feature maps with a size of 81x45 (32@81x45). And so on until the last layer, namely 128 feature vectors, is passed to the dense layer and sigmoid activation to produce a similarity value between two smartphone photo inputs. We utilized TensorFlow and keras libraries [33] to implement the whole model tasks, training, and classification. The number of CNN model parameters can be inferred from the architecture. Then we train Siamese with 2*s* images taken from two categories we compute the batch size. After repeating this epoch until loss is below some threshold, the resulting parameters of the network are then used to test the other 114-2s samples.

III. Results

From each of the two classes, we utilized 6 labelled samples each to perform multiple runs K times for cross-validation. For every single run, the training samples are replaced to perform the next **run**. In this scenario, the number of samples per class (shot) is 12/K for a 2-way *s*-shot. The combination of these multiple run *K* and shots, total train samples, and total test samples used is given in Table 2 below.

Table 2. Multiple runs K and shot number s

No	*K*	Num of classes	*s per class*	Total train	Total test
1	6	2	1	2	112
2	3	2	2	4	110
3	2	2	3	6	108
4	1	2	4	8	106
5	1	2	5	10	104
6	1	2	6	12	102

Parameters for both models are trained from the very first step without using any model adaptation or transfer learning. In this case, the support set is exactly the 2s train samples themselves, and the query set is the 114-2s test samples [34]. Since 2s samples are used for training, we tested the remaining 114 2*s* photos from both categories, student answer sheets, and fingerprint boards. The performance results of the scenario in terms of precision, recall, and F1 score [35], [36], [37] are presented in Table 3. In Table 3, when multiple run are made, K is bigger than 1, then the metrics in the cell are average values. The graphical view of Table 3 is shown in Fig. 6. The figure plots performance values on the vertical axis, while s-shot is the number of train samples per class on the horizontal axis.

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

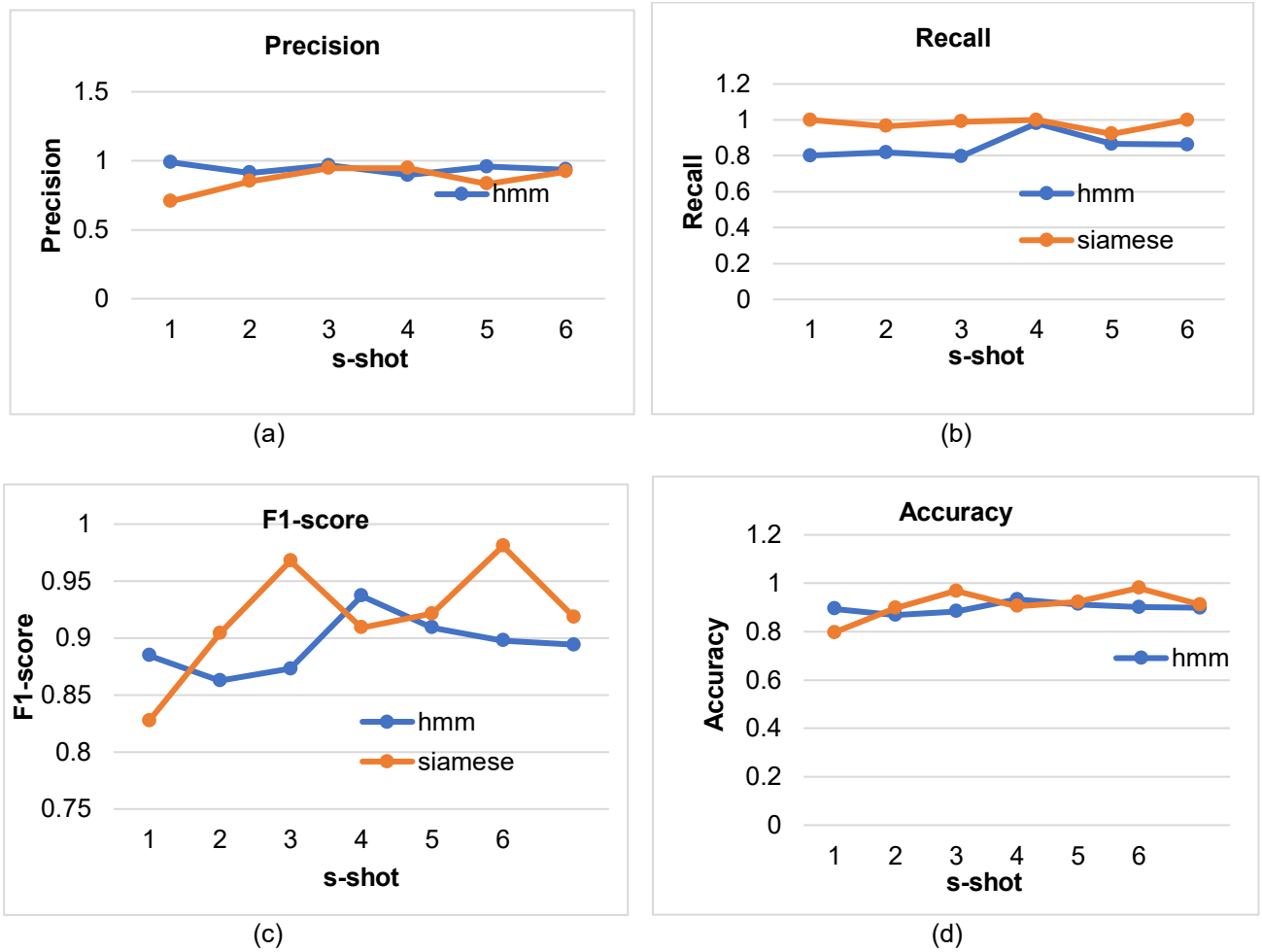


Fig. 6 Performance metrics (a) Precision, (b Recall), (c) F1-Score, (d) Accuracy

Table 3. Performance metrics: Precision, Recall and F1 score

		Precision		Recall		F1 score		Accuracy	
K	s	hmm	siamese	hmm	Siamese	hmm	siamese	hmm	siamese
6	1	0.989	0.706	0.801	0.999	0.885	0.827	0.895	0.797
3	2	0.912	0.851	0.818	0.965	0.863	0.904	0.869	0.897
2	3	0.966	0.946	0.796	0.991	0.873	0.968	0.883	0.968
1	4	0.897	0.946	0.981	1	0.937	0.909	0.933	0.906
1	5	0.957	0.833	0.865	0.922	0.909	0.922	0.913	0.923
1	6	0.936	0.922	0.863	1	0.898	0.981	0.901	0.98
Average		**0.943**	**0.867**	**0.854**	**0.979**	**0.894**	**0.918**	**0.899**	**0.912**

IV. Discussions

The study aims to develop classification models for a photograph taken with a smartphone, with the hope of helping people organize these photos quickly and accurately. This is because manual classification performed by humans is tedious and error-prone. The two models were built using Machine Learning algorithms, Hidden Markov and Siamese Convolutional Neural Network (CNN), with a focus on comparing the accuracy and computation speed of the two methods. This study used a photograph dataset, obtained from a personal phone of a lecturer at the university. This dataset consists of 114 observations and 2 categories. The dataset is already balanced with a 50 percent proportion for each category. Our dataset can be categorized as “simple” by the terms used in [34]. This study provides a baseline

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

performance of HMM methods versus Siamese CNN on a custom dataset. This research is based on the literature. The Siamese CNN in this study is based on [14]. The author in [14] used a Siamese CNN to categorize handwritten alphabets, Omniglot [38], which resulted in an accuracy of 92.0. This is the runner-up for the best results at that time, which was achieved by another method called Hierarchical Bayesian Program Learning (HBPL) with an accuracy of 95.2 [39]. This comparison is shown in Table 4.

Two methods used in this study are HMM and Siamese CNN. From the results of Fig. 6 and Table. 3 shows HMM still competes with the Siamese with very small differences. Fig. 6 and Table 3 show that the values for Precisions, Recalls, F1-Scores, and Accuracies are greater than 80% for both methods. This implies the successful classification using very few samples, with the number of samples ranging from one to six. The maximum ratio of the number of trains to test samples is 12/102, which is 11,8 percent. In this few-shot classification, we witness many times in the performance plot, the crossings of result lines between two methods. These crossings happen because of the change of the shot number, which drives the chance of metrics results in precision, f1-score,

Table 4. Comparison to previous studies for few-shot image classification

No	Method	*Datasets*	Accuracy
1	Siamese CNN [14]	Omniglot	92.2
2	HBPL [14]	Omniglot	95.2
3	HMM (this study)	This study	89.9
4	Siamese CNN (this study)	This study	91.2

and accuracy plots. At 1-shot, the precision of Siamese is 0.7, falling below HMM, which is 0.99. Then, by the increase of s to two samples, the Siamese takes over. The instability that showed an unsteady increase could happen because of the number of multiple runs K, which is for s=4,5,6, only 1-fold. We suggest that to achieve a more stable line in metrics, the number of multiple runs K should be increased. But increasing K means will require us to add more manual hand labelling, which is needed by the HMM method. This hand label is a drawback of the HMM method, in contrast to the Siamese, which does not need it. Performing multiple runs K-fold simulation in Siamese is simple procedure.

Overall performance can be judged from Table 3 and Fig. 6. All values are above 0.5, with the smallest value being 0.706 for the Precision of the Siamese method when only one train sample per category is used. Even Recall values from the Siamese method happen to reach perfection at 100% when the number of train samples is increased to four and six samples per category. The results also show an agreement and no contradiction between sensitivity and specificity. Since the ratio between positive and negative classes used in this experiment is roughly balanced, it is sufficient to judge the model based on the accuracy only [40], [41]. But what we witness here is that precision, recall, and f1-score also show the same trends for variation of the number of training samples, hence strengthening the validity of testing.

The computation speed of the HMM classifier should be highlighted here, as we suggest that this happens for a few reasons. This is the fact that HMM with no GPU added to the computer can still perform classification, which outputs results of adequate performance values. Firstly, the fewer computations of HMM are mostly because of its much lower number of parameters compared to the parameters of Siamese. Comparison between the number of parameters of HMM versus Siamese is in Table 5. The second reason is the path constraint that is applied in terms of grammar and dictionary by the HTK Toolkit. These two items significantly decrease the search space of the Viterbi algorithm used by HMM to perform classification. Incorporation of these aspects into the HMM classifier in the end will result in faster computation speed. While stepping back in the estimation algorithm, the Baum-Welch algorithm is very well established and optimized for decades, and in this research, it is proven again by showing its effectiveness. On the other side, the computation time needed by CNN families, including Siamese, is sometimes becoming a dilemma. While the parameter estimation process needs to be completed sooner, in fact, the devices needed to perform computation still need to be acquired at remarkable prices.

Table 5. Parameter Number of 2-way s-shot

No	Method	Number of parameters
1	HMM	2.205
2	Siamese	1.290.017

The conversion of grayscale before feature extraction could be a limitation of the HMM method applied in this case. As we know, we humans are very sensitive to color when classifying images. The DCT feature extraction has been proven in performance metrics to be effective. We deduce that this happens because of the DCT already represents low-level features, including shape, texture, and salient points of images [42]. But the color contribution to classification is still less. In performing photo recognition, if we want to mimic human intelligence, then the next research should improve the feature extraction technique by accommodating color channels information. In contrast, in this study, this color information has already been exploited by the Siamese method. This could be one reason for the superiority of Siamese over HMM. The effect of this study in real-world implications is the enabling of photo classification in low-performance devices where extra GPU devices are not present. This can be achieved by using HMM as a classifier.

V. Conclusion

This research aims to examine the HMM method for classification of photos taken with a smartphone. The examination is done by comparing the results of HMM against the Siamese CNN applied on the identical test and

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

training data split. CNN. From performance results, HMM can compete with results from a Siamese CNN. HMM achieves 0,94 precision, 0.85 recall, F1 score 0.89, and accuracy 0.90 while Siamese claims 0.87, 0.98, 0.92, and 0.91. The result shows that HMM, which has an advantage over Siamese in terms of fewer parameters, still challenges Siamese CNN with only a slight decrease in performance. We conclude that HMMs are suitable over Siamese CNNs to be implemented in low-performance devices such as cellphones. Future research on HMMs apart from the current setup is to improve preprocessing, feature extraction, use more complex state topologies, and adaptation and testing to larger datasets. The dataset and source code for this research are available from https://github.com/dzhatala/hmmphoto.

Declarations

Consent for Publication Participants.

Consent for publication was given by all participants

Competing Interests

The authors declare no competing interests.

References

[1] B. Mor, S. Garhwal, and A. Kumar, "A Systematic Review of Hidden Markov Models and Their Applications," *Arch Computat Methods Eng*, vol. 28, no. 3, pp. 1429–1448, May 2021, doi: 10.1007/s11831-020-09422-4.

[2] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *PROCEEDINGS OF THE IEEE*, vol. 77, no. 2, 1989.

[3] Y. Ephraim and N. Merhav, "Hidden markov processes," *IEEE Transactions on information theory*, vol. 48, no. 6, pp. 1518–1569, 2002.

[4] F. Bimbot *et al.*, "A tutorial on text-independent speaker verification," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 4, p. 101962, 2004.

[5] D. Ali, I. Touqir, A. M. Siddiqui, J. Malik, and M. Imran, "Face Recognition System Based on Four State Hidden Markov Model," *IEEE Access*, vol. 10, pp. 74436–74448, 2022, doi: 10.1109/ACCESS.2022.3188717.

[6] A. Aggarwal, M. Alshehri, M. Kumar, P. Sharma, O. Alfarraj, and V. Deep, "Principal component analysis, hidden Markov model, and artificial neural network inspired techniques to recognize faces," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 9, p. e6157, 2021.

[7] M. Mouret, C. Solnon, and C. Wolf, "Classification of Images Based on Hidden Markov Models," in *2009 Seventh International Workshop on Content-Based Multimedia Indexing*, Chania, Crete: IEEE, Jun. 2009, pp. 169–174. doi: 10.1109/CBMI.2009.22.

[8] C. Wang, Z. Yu, Z. Long, H. Zhao, and Z. Wang, "A few-shot diabetes foot ulcer image classification method based on deep ResNet and transfer learning," *Scientific Reports*, vol. 14, no. 1, pp. 1–9, 2024.

[9] J. Dong, Y. Wang, J.-H. Lai, and X. Xie, "Improving adversarially robust few-shot image classification with generalizable representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9025–9034.

[10] B. M. S. Maia *et al.*, "Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer," 2024.

[11] K. Dunphy, M. Buwaneswaran, K. Grolinger, and A. Sadhu, "Few-Shot Learning Augmented with Image Transformation for Multiclass Structural Damage Classification," *Journal of Computing in Civil Engineering*, vol. 39, no. 3, p. 04025021, 2025.

[12] W. Song and Y. Huang, "Adaptive feature recalibration transformer for enhancing few-shot image classification," *The Visual Computer*, pp. 1–15, 2025.

[13] J. Ridha, K. Saddami, M. Riswan, R. Roslidar, and others, "An Explainable Artificial Intelligence Framework for Breast Cancer Detection," *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 7, no. 2, pp. 298–311, 2025.

[14] G. Koch, R. Zemel, R. Salakhutdinov, and others, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, 2015.

[15] M. Hamid, "DCT-based image feature extraction and its application in image self-recovery and image watermarking," PhD Thesis, Concordia University, 2016.

[16] L. Tan and J. Jiang, *Digital signal processing: fundamentals and applications*, 3rd ed. Academic press, 2019.

[17] V. V. Kohir and U. B. Desai, "Face recognition using a DCT-HMM approach," in *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*, Princeton, NJ, USA: IEEE Comput. Soc, 1998, pp. 226–231. doi: 10.1109/ACV.1998.732884.

[18] J. Li and R. M. Gray, *Image segmentation and compression using hidden Markov models*, vol. 571. Springer Science & Business Media, 2000.

[19] F. S. Samaria, "Face recognition using hidden Markov models," PhD Thesis, University of Cambridge Cambridge, UK, 1994.

[20] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden Markov model," *IEEE transactions on signal processing*, vol. 48, no. 2, pp. 517–533, 2000.

[21] S. Osaki, *Applied stochastic system modeling*. Springer Science & Business Media, 2012.

[22] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, Fourth. Boston: McGraw Hill, 2002. [Online]. Available: http://www.worldcat.org/search?qt=worldcat_org_all&q=0071226613

[23] S. Ross, *First Course in Probability, A: Pearson New International Edition PDF eBook*. Pearson Higher Ed, 2013.

[24] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang, "A segmental k-means training procedure for

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

connected word recognition," *AT&T technical journal*, vol. 65, no. 3, pp. 21–31, 1986.
[25] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
[26] A. Allahverdyan and A. Galstyan, "Comparative analysis of viterbi training and maximum likelihood estimation for hmms," *Advances in neural information processing systems*, vol. 24, 2011.
[27] S. J. Young, N. H. Russell, and J. H. S. Thornton, "Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems".
[28] R. Solera-Ureña, J. Padrell-Sendra, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, and F. Díaz-de-María, "SVMs for Automatic Speech Recognition: A Survey," in *Progress in Nonlinear Speech Processing*, vol. 4391, Y. Stylianou, M. Faundez-Zanuy, and A. Esposito, Eds., in Lecture Notes in Computer Science, vol. 4391. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 190–216. doi: 10.1007/978-3-540-71505-4_11.
[29] H. Shao, D. Zhong, X. Du, S. Du, and R. N. Veldhuis, "Few-shot learning for palmprint recognition via meta-siamese network," *IEEE transactions on instrumentation and measurement*, vol. 70, pp. 1–12, 2021.
[30] Y. Lai, "A comparison of traditional machine learning and deep learning in image recognition," in *Journal of Physics: Conference Series*, IOP Publishing, 2019, p. 012148.
[31] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on signal processing*, vol. 65, no. 13, pp. 3551–3582, 2017.
[32] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1845–1866, 2017.
[33] O. Dürr, B. Sick, and E. Murina, *Probabilistic deep learning: With python, keras and tensorflow probability*. Manning Publications, 2020.
[34] Y. Liu, H. Zhang, W. Zhang, G. Lu, Q. Tian, and N. Ling, "Few-shot image classification: Current status and research trends," *Electronics*, vol. 11, no. 11, p. 1752, 2022.
[35] L. Siena, T. H. Saragih, R. A. Nugroho, D. Kartini, W. Caesarendra, and others, "Evaluation of the Impact of SMOTEENN on Monkeypox Case Classification Performance Using Boosting Algorithms," *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 7, no. 2, pp. 203–220, 2025.
[36] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.
[37] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers & Operations Research*, vol. 152, p. 106131, 2023.
[38] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.
[39] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," *Advances in neural information processing systems*, vol. 26, 2013.
[40] E. A. Freeman and G. G. Moisen, "A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa," *Ecological modelling*, vol. 217, no. 1–2, pp. 48–58, 2008.
[41] A. H. Fielding and J. F. Bell, "A review of methods for the assessment of prediction errors in conservation presence/absence models," *Environmental conservation*, vol. 24, no. 1, pp. 38–49, 1997.
[42] J. Yue, Z. Li, L. Liu, and Z. Fu, "Content-based image retrieval using color and texture fused features," *Mathematical and Computer Modelling*, vol. 54, no. 3–4, pp. 1121–1127, 2011.

Author Biography

Zulkarnaen Hatala received his B.CS. Degree in Informatics, his M.Eng. Degree in Electrical Engineering, both from Telkom University Bandung, Indonesia formerly called Sekolah Tinggi Teknologi Telkom. His academic and research interests lie in the fields of machine learning, computer vision, and image processing, particularly in the application of deep learning techniques for solving real-world problems. Throughout his studies, he has been actively involved in various research projects and academic activities related to artificial intelligence and its applications in engineering. Currently, he serves as a lecturer at the Informatics Study Program, Politeknik Negeri Ambon (Polnam), Ambon, Indonesia, where he continues to teach and conduct research. He can be contacted at email: dzulqarnaenhatala@gmail.com.

Muhammad Hudzaly received his Bachelor's Degree in Information Systems from Universitas Islam Negeri Alauddin Makassar and his Master's degree in Industrial Engineering and Management from Universitas Diponegoro. His academic and research interests include preventive maintenance, safety management, downtime analysis, and the use of artificial intelligence technologies to support decision-making in industry. With a background in industrial engineering, he is committed to advancing knowledge and practice that enhance efficiency, safety, and sustainability within industrial

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116

environments using technology. For now, he serves as a Lecturer at the Department of Industrial Engineering, Institut Teknologi Kalimantan (ITK), Balikpapan, Indonesia, where he is committed to teaching and conducting research. He can be contacted via email: m.hudzaly@lecturer.itk.ac.id.

Corresponding author: Zulkarnaen Hatala, dzulqarnaenhatala@gmail.com, Teknik Informatika Politeknik Negeri Ambon, Ambon, Indonesia
Digital Object Identifier (DOI): https://doi.org/10.35882/ijeeemi.v7i3.116