Scientific Research Publishing

Journal of Modern Physics, 2021, 12, 361-379
https://www.scirp.org/journal/jmp
ISSN Online: 2153-120X
ISSN Print: 2153-1196

Video-Based Face Recognition with New Classifiers

Soniya Singhal[1], Madasu Hanmandlu[2], Shantaram Vasikarla[3]

[1]Electrical Engineering Department, Indian Institute of Technology, New Delhi, India
[2]CSE Department, MVSR Engg. Cllege, Nadergul, Hyderabad, India
[3]Computer Science Department, California State University, Northridge, CA, USA
Email: soniyatech90@gmail.com, mhmandlu@gmail.com, shantaram@computer.org

How to cite this paper: Singhal, S., Hanmandlu, M. and Vasikarla, S. (2021) Video-Based Face Recognition with New Classifiers. *Journal of Modern Physics*, **12**, 361-379.
https://doi.org/10.4236/jmp.2021.123026

Received: January 14, 2021
Accepted: February 23, 2021
Published: February 26, 2021

Abstract

An exhaustive study has been conducted on face videos from YouTube video dataset for real time face recognition using the features from deep learning architectures and also the information set features. Our objective is to cash in on a plethora of deep learning architectures and information set features. The deep learning architectures dig in features from several layers of convolution and max-pooling layers though a placement of these layers is architecture dependent. On the other hand, the information set features depend on the entropy function for the generation of features. A comparative study of deep learning and information set features is made using the well-known classifiers in addition to developing Constrained Hanman Transform (CHT) and Weighted Hanman Transform (WHT) classifiers. It is demonstrated that information set features and deep learning features have comparable performance. However, sigmoid-based information set features using the new classifiers are found to outperform MobileNet features.

Keywords

Face Recognition on Videos, Information Sets, Constrained Hanman Transform Classifier, Weighted Hanman Transform Classifier, Video Face Dataset, MobileNet, Vgg-16, Inception Net, ResNet

1. Introduction

Face recognition from videos is still a challenging problem. One source of videos is CCT cameras installed wherever the thefts and criminal activities are expected. We can see the use of these cameras for surveillance and security at places like airports, hospitals, banks, streets, homes, offices etc. With the increased terrorist

attacks, concern about public safety has become paramount importance worldwide. Earlier the passport photo used to be the only identity for travelers, but now secrete cameras at airports keep a strict vigil on their activities as well. They record real time videos during the immigration checks. Their applications are furthered in detecting the poses of people, tracking the objects and activities.

Face recognition has been in vogue for the user verification and authentication, public safety, attendance management and counting of people. But real time detection and recognition of a face are very difficult as they have to be faster without compromising on accuracy. Tracking the presence of an authorized human is an added overhead to prevent his/her access.

There are two approaches based on 1) image-based and 2) video-based. The first approach has been researched from multiple perspectives, not limited to performance, computational constraints, and image acquisition under both the constrained and unconstrained environments including occlusions. On the other hand, the second approach has not been explored to that extent desirable and it is still an active research field due to an additional challenge posed by videos. Motivated to counter this challenge, we have made an attempt to investigate both the information set features and deep learning neural network features in this work. Moreover, depending on an application, video based face recognition can be performed using either image-to-video or video-to-video. The image-to-video methods are suited to applications involving an identification of a person. Here the presence of a person in videos is determined based on still image dataset. On the other hand, video-to-video matching is mainly used to find all the occurrences of a subject within a collection of video data. Here both the system input and the database are in the form of videos and video-to-video matching is more challenging than image-to-video matching. Typical solutions to this problem involve multiple stages like extracting features from the input videos and then matching with the target video features. In this work, we are concerned with the real time face recognition involving image-to-video methods.

There are multiple factors in the way of performing real time face recognition. Pose and location of a person in a video vary widely. Moreover, the varying expressions, illumination, background, occlusions affect the processing after the acquisition of the video. Every frame of a face video may witness a significant change in the form of pose, expression and illumination. Considering all the frames will make a face recognition system slow because of computational complexity. Therefore, only a few frames at a certain frame rate have to be selected to reduce the training time in the recognition system.

Zheng *et al.* [1] have proposed an automatic system for the unconstrained video-based face recognition. First of all, faces are localized in videos using two Deep Convolutional Neural Networks (DCNN) detectors. The regions of bounding boxes earmarked from the earlier step are then grouped based on face association and tracking methods. Faces are then recognized by the face matcher using an unsupervised subspace learning approach and a subspace-to-subspace

similarity metric.

Deep learning networks have been favored by researchers in the field of image processing as they are well suited for the extraction of features from images of all sorts and classifying the same by the inbuilt classifiers like softmax in the very networks. Face recognition, emotion recognition, optical character recognition, detection of diseases from medical images are some of the applications where deep learning networks have become a natural choice. They outperform the classical methods for the pattern recognition and classification problems by extents. Because of their wide usage, several deep learning networks have emerged starting with convolutional neural networks (CNN) having different architectures like Alex Net, LeNet, Vgg-16 to Region-CNN (RCNN), MobileNet, ResNet etc.

A survey on face recognition using deep learning is made in [2] on two aspects: data and algorithms. For algorithms, network architectures are surveyed. Also, they have categorized loss functions into Euclidean-distance based, angular or cosine-margin based, softmax and its variations. For data, some commonly used datasets are surveyed. As mentioned in [2] that technical challenge with matching faces cross ages, poses and styles still remains. Deep learning has improved continuously over the past few years and now it even assists humans in face verification. But the applications requiring high accuracy at a very low alarm rate like financial identity verification are still difficult even with massive training data.

We will discuss some of the deep learning neural networks later on in connection with our application. They are not without drawbacks as can be noticed. Training these models is cumbersome as they need huge dataset to train from scratch; but with transfer learning the need for huge training data can be reduced. As there are many deep learning architectures in the literature the choice and suitability of particular architecture to an application are fraught with difficulties. The unavailability of big public databases hampers the effective use of deep learning neural networks. To mitigate this problem, data augmentation from the available samples is employed during training. Traditional methods seek image synthesis for augmentation that includes but not limited to cropping, rotating, flipping, random sampling or adding noise in images and these operations involve transformations that do not affect the category. In real life scenario, the augmentation may not be necessitated but cannot altogether be ruled out.

Close on the heels of real life scenario, Inoue [3] has suggested an augmentation technique that takes the average of two randomly selected images from a training set. If two labeled images are chosen, then the label of the first is considered for the generated one. This method of sample pairing leads to reduction in error rate over the traditional approaches. But the selection of images has a significant impact on the accuracy. Surprisingly, if images belong to the same class, improvement is poorer. There is no valid reason for this anomaly therefore choosing sample pairs is a difficult task. Generative Adversarial Nets (GAN) [4] can generate new images similar in all aspects to the training images. They com-

prise two networks, namely, generator and discriminator that compete with each other in performance. The generator tries to make similar input images while the discriminator estimates the probability of whether a sample is taken from the original network and not from the generated network. The learning aim of generator is to increase the probability of discriminator making a mistake. It is interesting to train GANs if the number of training set images is small. Also, GAN requires the choice of a good equilibrium which is not ensured thus leading to unstable training.

We are more interested in the transfer learning approaches for Deep Neural Networks (DNNs) as they save time to build a model not from scratch. Moreover, this sort of learning helps solve other's problem. DNN's can deal with small training datasets. Lin *et al.* [5] have proposed a transferred deep feature fusion framework that utilizes two Deep Convolution Neural Networks (DCNNs) for feature extraction and identification of the unknown (test) face from the known (training) faces. The architectures used are ResNet-50 and GoogLeNet-BN and these are trained on different databases to get more generalized feature representation. For training, data augmentation is adopted by flipping, cropping and resizing images. The fusion occurs at two stages, one at the features level and another at the similarity scores level. In the first fusion stage, the features extracted from the output layers of the networks are fused together. Then template based on one vs. the rest—SVM is trained on these features. A template is a collection of all frames of a video and the images of a focused subject are used as a single representation for the matching task. Finally, One Shot Similarity (OSS) is applied to identify the input. OSS of two vectors say p and q, is measured by considering p as a positive set and all other samples that don't belong to either of these two vectors as a negative set and then classification is performed. Similarly, it's done for q and then the average of these is taken. In the second fusion stage, multiple matching scores obtained for each template-pair are fused into a single score.

Most of the research works on DCNNs [1] [2] [4] [5] have some common problems. They require larger dataset, memory and computation time. As we are all aware bigger the dataset the better is the model and so the need arises for data augmentation. Though this helps reduce the problem of overfitting but is computationally expensive and also increases the training time. Sample pairing type augmentations [3] have a disadvantage that they make a little sense from the humanistic perspective. It is very difficult to interpret why the performance boost takes place by mixing images. One reason could be that the increase in the data size facilitates robust representation of the low-level characteristics. For Generative Adversarial Networks (GANs) in [4], getting high-resolution output is still challenging. Increasing the output size of the images produced by the generator is likely to cause training instability and non-convergence. Moreover, fine tuning of these networks is also challenging. Another drawback is that they do not encode the position and orientation of the objects.

The present work involving real time face recognition using videos is in continuation of our earlier work on face recognition [6] where we have addressed the real time face recognition problems such as pose and illumination variations using the information set features that arise out of the representation of the possibilistic certainty using Mamta-Hanman entropy function. To see the relative merits of information set features vis-a-vis deep learning-based features, we are motivated to investigate the effectiveness of DCNNs on videos by overriding their requirement for large databases with smaller databases. As there are many candidates to choose from different deep learning neural architectures, some experimental study will be conducted. We are also bent upon developing two classifiers, viz., Constrained Hanman transform (CHT) classifier and Weighted Hanman transform (WHT) classifier in this work.

The rest of the paper is organized as follows: Section 2 presents the extraction of features. Section 3 contains the proposed classifier and algorithm. DNNs used are discussed in Section 4. The details of the databases are given in Section 5. Section 6 discusses the results of experiments on two video databases. The conclusions are given in Section 7.

2. The Concept of Information Set

The concept of information set was introduced by Hanmandlu in a guest editorial [7] so as to extend the scope of a fuzzy set by empowering the membership function involved in it as an agent through the proposition of an information theoretic entropy function christened as Hanman-Anirban entropy function [8]. This is more general in the sense that it can represent both probabilistic uncertainty and the possibilistic certainty unlike Shannon, Renyi, Tsallis and Pal and Pal entropy functions that represent only the probabilistic uncertainty considered as a measure of disorder called information. The offshoot of Hanman-Anirban is called Mamta-Hanman entropy function [9], defined as

$$H = \frac{1}{n^2} \sum \sum p_{ij}^{\alpha} \mathrm{e}^{-\left(cp_{ij}^{\gamma}+d\right)^{\beta}} \tag{1}$$

where c, d, α, β and γ are the constant parameters and p_{ij} is the probability. For simplicity we take $\gamma = 1$ in Equation (1) that represents the probabilistic uncertainty. In this paper, we are concerned with the possibilistic certainty in the gray levels of a sub-image which is proved to be better than the probabilistic uncertainty. We term a sub-image as the information source and the gray levels as the information source values to prepare the ground to represent the possibilistic certainty as will be clear shortly. To this end, we replace the probability p_{ij} with the information source values I_{ij} in Equation (1) leading to:

$$H = \frac{1}{n^2} \sum \sum I_{ij}^{\alpha} \mathrm{e}^{-\left(cI_{ij}^{\gamma}+d\right)^{\beta}} \tag{2}$$

We now choose the parameters of the exponential gain function such that it takes the form of a membership function whose role is to fit an appropriate pos-

sibility distribution for the information source values.

Selecting $\gamma = 1$, $c = \frac{1}{I_{\max}}$, $d = 1 - \frac{I_{avg}}{I_{\max}}$ in Equation (2) leads to

$$H = \frac{1}{n^2}\sum\sum I_{ij}^{\alpha} \mathrm{e}^{-\left(1-\mu_{ij}\right)^{\beta}} = \frac{1}{n^2}\sum\sum I_{ij}^{\alpha} \mathrm{e}^{-\bar{\mu}_{ij}^{\beta}} \tag{3}$$

where $0 < \alpha < 2$ and the membership function is $\mu_{ij} = \frac{\left|I_{ij} - I_{avg}\right|}{I_{\max}}$. Taking the first order approximation in the exponential gain function in Equation (3) leads to,

$$H = \frac{1}{n^2}\sum\sum I_{ij}^{\alpha}\left(1 - \bar{\mu}_{ij}^{\beta}\right) = \frac{1}{n^2}\sum\sum I_{ij}^{\alpha}\mu_{ij}^{\beta} \tag{4}$$

As we have modeled the distribution of I_{ij} to get μ_{ij}, Equation (4) gives the possibilistic certainty; but if we replace μ_{ij} with its complement then it gives the possibilistic uncertainty.

2.1. Dilemma between Certainty and Uncertainty

It is time to understand the difference between the two. We have been making a concerted effort over the years through the aegis of information set theory to clear up this dilemma. As probabilities express a random phenomenon, the classical entropy functions give a measure of disorder or uncertainty in a system. In some situations probabilities act as possibilities. For instance, the occurrences of minerals in sea water are random, but their effect on the taste of the water is not random but fuzzy as we can easily associate a concept like bitterness to it because the variation in bitterness leads to a fuzzy set. The degree of bitterness is described by a membership function which gives a certainty value. But the bitterness depends on the amounts of minerals dissolved in the water.

The variation in any information source (attribute) values gives rise to a distribution. If this distribution can be modelled by a mathematical function using the statistics of the distribution termed as the possibilistic entropy function, then we get the extent of certainty of the attribute to a specified concept or class. Otherwise the distribution is random; hence it can only be represented by a mathematical function without using the statistics of the distribution termed as the probabilistic entropy function in which case we get the extent of uncertainty involved in relating the variable to the concept/class. The difference between the two is whether or not the distribution of a variable can be modelled using its statistics. This is termed as certainty/uncertainty principle.

The advantage of using statistical parameters in the modelling of a distribution of information source values by a mathematical function bestows us the facility to change the parameters thereby changing the function. If the information source values have the corresponding degrees of association provided by a mathematical function to a concept or class then both the information source values and the degrees of association together represent the certainty whether or

not the mathematical function involves the statistical of parameters of the distribution of the information source values. We will be using this simple concept while designing the classifiers. We now define the information set concept that is the backbone of the information set theory.

2.2. Definition of Information Set

The set of information values $\left\{I_{ij}^{\alpha}\mu_{ij}^{\beta}\right\}$ is called the information set such that each information value is a product of the information value and the corresponding membership function value. The values of α and β need to be selected appropriately.

An in depth of study on information set theory can be found from [10] [11] [12]. The adaptive forms of Mamta-Hanman and Hanman-Anirban entropy functions are presented in [6] and [12] respectively, which will give more teeth to be able to derive Hanman transform from them as shown later.

2.3. Operations on Information Sets

Let H1 and H2 be two information sets. The operations of union, Intersection and complement are given as under.

1) *Union*: It is the t-norm of $H1_{ij}$ and $H2_{ij}$ that are the corresponding information values of two information sets.

2) *Intersection*: It is the t-conorm or S-norm of $H1_{ij}$ and $H2_{ij}$.

3) *Complement Information*: If the membership function is complement we get complement information $\left\{I_{ij}^{\alpha}\overline{\mu}_{ij}^{\beta}\right\}$.

4) *Thresholding*: Unlike cut-sets, the information sets are subjected to thresholding for the choice of information content.

5) *Functions*: Information set allows generation of modified features by applying different functions on the basic information values.

2.4. Functional Information Set Features

To derive these features, the unit of information is taken as either the information value, $I_{ij}^{\alpha}\mu_{ij}^{\beta}$ or the complement information value, $I_{ij}^{\alpha}\overline{\mu}_{ij}^{\beta}$ and then an appropriate function is applied on this unit information. The formulation of such features is now discussed.

2.4.1. Energy (EN) Feature

Energy feature is derived from Equation (4) by taking $\beta = 2$ in $\overline{\mu}_{ij}^{\beta}$ for the k^{th} window denoted by E_k as:

$$E_k = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} I_{ij}^{\alpha}\overline{\mu}_{ij}^{2} \tag{5}$$

2.4.2. Sigmoid (SG) Feature

Applying the sigmoid function on the unit of information leads to the following sigmoid feature (SG) denoted by S_k:

$$S_k = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{I_{avg}}{1+\mathrm{e}^{-I_{ij}^{\alpha}\mu_{ij}^{\beta}}} \tag{6}$$

2.4.3. Effective Information (EI) Feature

As the above μ_{ij}^{β} is not found suitable, we have chosen an exponential membership function $\mu_{ij} = \mathrm{e}^{-\left(\frac{\left|I_{ij}-I_{ref}\right|}{f_h^2}\right)}$ which is obtained from the exponential gain function $\mathrm{e}^{-\left(cI_{ij}^{\gamma}+d\right)^{\beta}}$ of (2) on substituting $\gamma = 1$, $c = \frac{1}{f_h^2}$, $d = -\frac{I_{ref}}{f_h^2}$. The use of centroidal approach on the information values $I_{ij}^{\alpha}\mu_{ij}^{\beta}$ gives the effective information (EI) denoted by I_k as follows:

$$I_k = \frac{\sum_i\sum_j I_{ij}^{\alpha}\mu_{ij}^{\beta}}{\sum_i\sum_j \mu_{ij}^{\beta}} \tag{7}$$

where I_{ref} can be taken as $I_{\max}$ and the fuzzifier is given by:

$$f_h^2 = \frac{\sum_i\sum_j\left(I_{ij}-I_{ref}\right)^4}{\sum_i\sum_j\left(I_{ij}-I_{ref}\right)^2}$$

2.4.4. Possibilistic Renyi Entropy (RE) Feature

Renyi entropy function is based on probabilities; so it cannot be used as feature as we are dealing with the attribute (information source) values. But its possibilistic form is derived by Bhatia and Hanmandlu in [13] by making it adaptive. We will first consider the Renyi entropy (RE) function with the probability replaced by the information source value. Denoting RE by R_k for k^{th} window, it is expressed as

$$R_k = \frac{1}{n^2}\frac{1}{1-\alpha}\left\{\log\left(\sum_{i=1}^{n}\sum_{j=1}^{n}I_{ij}^{\alpha}\right)\right\} \tag{8}$$

Since $\left(\sum_{i=1}^{n}\sum_{j=1}^{n}I_{ij}\right)\neq 1$, unlike the sum of the probabilities equals 1 we have normalized the r.h.s. of (8) by dividing with the number of the information source values I_{ij} to get what we call the approximate normalized possibilistic Renyi entropy function. We have fixed $\alpha = 2$ obtained by experimentation as discussed in Section 6 on results; so RE feature is computed from

$$R_k = \frac{1}{n^2}\left\{\log_{\mathrm{e}}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}I_{ij}^{2}\right)\right\}.$$

We will now derive adaptive Renyi entropy (ARE) function by assuming α to be a variable.

Then one term of (8) becomes:

$$R_{A,ij} = \frac{\alpha}{1-\alpha}\left\{\log_{\mathrm{e}} I_{ij}\right\} \tag{9}$$

Assuming $\frac{1}{1-\alpha} = \mu_{ij}$ makes $\frac{\alpha}{1-\alpha} = -\bar{\mu}_{ij}$, the complementary membership function of I_{ij}. In view of this (9) becomes

$$R_{A,ij} = -\bar{\mu}_{ij}\left\{\log_e I_{ij}\right\} \tag{10}$$

where $\mu_{ij} = \dfrac{\left|I_{ij} - I_{avg}\right|}{I_{\max}}$.

Summing the above for $i = 1,\cdots,n$ and $j = 1,\cdots,n$ we get ARE as:

$$R_{Ak} = -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\bar{\mu}_{ij}\log_e I_{ij} \tag{11}$$

2.5. Derivation of Hanman Classifier

A classifier works on the training feature vectors and a testing feature for the classification we denote a feature vector by $\{x_i\}$. We will now derive Hanman transform that is a higher level information set. To derive this, we invoke the adaptive Mamta-Hanman entropy function [6], defined as

$$H = \frac{1}{n}\sum x_i^{\alpha} e^{-\left(c(\,)x_i^{\gamma} + d(\,)\right)^{\beta}} \tag{12}$$

where the parameters $c(\,)$ and $d(\,)$ are variables and x_i^{α} denotes the feature vector. Setting the parameter $c(\,)$ to μ_i and $d(\,)$ to zero in Equation (10) yields us the most general Hanman Transform, given by

$$H = \frac{1}{n}\sum x_i^{\alpha} e^{-\left(\mu_i x_i^{\gamma}\right)^{\beta}} \tag{13}$$

To simplify the above, the three parameters: α, γ and β are set to unity resulting in the basic Hanman transform:

$$H = \frac{1}{n}\sum x_i e^{-(\mu_i x_i)} \tag{14}$$

The above requires the generation of feature vector x_i and its membership function. However, it would be more effective if we compute the error vector between the training feature vector and the test feature vector. The development of classifier based on this transform will be discussed now.

2.6. Properties of Information Sets and Hanman Transform

To give an insight into the Information set theory, it is essential to enlist some important properties some of which are common to both information sets and Hanman transform and some specific to either of the two. These are discussed in the following:

1) Both information values and Hanman transform values are natural variables. The electro-chemical pulse from dendrite is either magnified or inhabited by the synapse before reaching a neuron. This is equivalent to changing an attribute by its membership value as in a basic information value. Evaluation of an attribute based on the information on it gives the Hanman transform value and the function of this transform is similar to the higher level activity of neurons in the human brain.

2) The summation of information values gives us an estimate of the output.

This useful result helps us simplify the Takagi-Sugeno-Kang (TSK) fuzzy rule.

Proof. Consider a TSK fuzzy rule

If x_1 is A_1 and x_2 is A_2 and x_n is A_n then $y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n$

If the fuzzy sets $A_1, \cdots, A_n$ are replaced with the information values, then we can write $y = \mu_1x_1 + \mu_2x_2 + \cdots + \mu_nx_n$. This output is a sum of information values with $a_0 = 0$.

3) From 2), we can deduce that $y = w_1\mu_1x_1 + w_2\mu_2x_2 + \cdots + w_n\mu_nx_n$ for the situation when $A_1, \cdots, A_n$ have different sizes.

4) Transfer learning is possible through Hanman transform $H = \frac{1}{n}\sum x_i e^{-(\mu_i(y)x_i)}$ where the membership $\mu_i(y)$ is derived from another attribute *y*.

5) If μ_1 and μ_2 are treated as two agents then $\{x(\mu_1 - \mu_2)\}$ is the divergent information set and the Hanman transform divergent set is simply the corresponding Hanman transform $\{x\exp(-x(\mu_1 - \mu_2))\}$.

6) Both basic information value and Hanman transform can be used in the learning of parameters in an optimization problem [14] [15] using $H_a = a_i f(a_i)$ and $H_a = \frac{1}{n}\sum a_i e^{-a_i f(a_i)}$ respectively. Here a_i is the parameter set to be learned and $f(a_i)$ is the output of the objective function.

7) Hanman transform can be used to evaluate the membership function value as $\mu_i(new) = \mu_i(old)e^{-x_i\mu_i(old)}$ which is recursive. If we have a correction term, say $v_i(old)$ necessitated due to inappropriate membership function model, then we can have a non-linear state space model from this recursive relation represented as: $\mu_i(k+1) = \mu_i(k)e^{-x_i\mu_i(k)} + v_i(k)$, where *k* denotes the k^{th} instant. The model parameters of this model can be easily learned using competitive-cooperative learning models (cclms) which also use the output-based Hanman transform [15] as mentioned in 6).

3. Design of Classifier Based on Hanman Transform

Unlike in many classifiers there is no training phase in the design of this classifier but only the testing phase because the need for the unknown parameters is eliminated completely by assuming suitable parameters in the Hanman transform. Separating training phase from testing phase would entail the computation of parameters which we have avoided by having only the testing phase. This approach is amply suitable for the case of availability of only a few training samples.

Here, we compute the error vectors between the training feature vectors and test feature vector and then taking two error vectors at a time we compute all the possible t-normed error vectors. From these normed error vectors, we select the one with the least entropy value thus eliminating all other normed error vectors. This acts as a support vector for the class.

Proof. As we are extracting information set value/feature from each window/sub-image and sum of these values over all windows or the entire image gives the

certainty because $H = \frac{1}{n}\sum x_i \mu_i$ which in the case of unsupervised learning provides an estimate of the output whereas $\bar{H} = \frac{1}{n}\sum x_i (1-\mu_i)$ gives uncertainty or disorder which we are not considering here. Subtraction of information values of the training image and test image leads to one error vector $\{e_i\}$. As we have several training images and one test image, subtraction of their information value gives several error vectors or a set of error vectors. The t-norm of any two error vectors yields the minimum of these vectors called t-normed error vector denoted by $\{\tilde{e}_i\}$. Let us compute the certainty value of this t-normed error vector as $E = \frac{1}{n}\sum \tilde{e}_i \tilde{\mu}_i(\tilde{e}_i)$, where the t-normed error vector $\{\tilde{e}_i\}$ is converted into membership function vector, $\{\tilde{\mu}_i(\tilde{e}_i)\}$ called M-error vector. If we select the t-normed error vector with the least certainty value out of all possible t-normed error vectors of a user, it means all other t-normed errors have more certainty than the one we have selected and thus it serves as the limit of tolerance which is akin to the support vector of SVM for a class/user. The information set theory offers us an easy way to determine the limit of tolerance as against lengthy computation required to compute the support vectors in support vector machine (SVM). In the proposed approach we will be using Hanman transform to compute the least certainty valued t-normed error vector for the high level representation of certainty as follows: $E_T = \frac{1}{n}\sum \tilde{e}_i \exp\{-\tilde{e}_i \tilde{\mu}_i(\tilde{e}_i)\}$. Here we have used the exponential membership function $\tilde{\mu}_i(\tilde{e}_i)$ of $\tilde{e}_i$ without using the statistical parameters. But their product gives the certaintyof t-normed error vector belonging to a class.

We will now present two classifiers based on Hanman transform (HT). The first classifier is called Constrained HT (CHT) for which an objective function J_M is formed using M-vector. The user with the least value of the product, ($E_T J_M$) is identified with the unknown user or test sample. The second classifier is called Weight HT (WHT) for which we form a weight vector using the values of M-vector and these weights are used in the Hanman transform to get the least value of $E_{WT} = \frac{1}{n}\sum w_i \tilde{e}_i \exp\{-\tilde{e}_i \tilde{\mu}_i(\tilde{e}_i)\}$ that gives the identity of the unknown user.

Before presenting the algorithm, let the length of feature vector be *n*, the number of training samples be *s* for each class, the number of test samples be *n'*andthe number of classes be *C*. It may be noted that we have used a different notation in the algorithms from the above for the convenience of representation.

3.1. Algorithm-1 of Constrained Hanman Transform (CHT) Classifier

1) Compute the error vector between the m^{th} training feature vector of l^{th} user x_{mj}^{l} and the test feature vector t_j as given by

$$e_{mj}^{l} = \left| x_{mj}^{l} - t_j \right| \tag{15}$$

where $m = 1, 2, \cdots, s$ and $j = 1, 2, \cdots, n$.

2) Compute the Frank t-normed error vector for I^{th} user on a pair (m,h) of the error vectors using

$$E_{mh}^{l}(j) = T\left(e_{mj}^{l}, e_{hj}^{l}\right); m \neq h \tag{16}$$

where $T(x, y) = \log_p\left(1 + \frac{\left(p^x - 1\right)\left(p^y - 1\right)}{p - 1}\right)$ and p is set to 2.

3) Compute the exponential membership function of t-normed error vector from:

$$M_{mh}^{l}(j) = \mathrm{e}^{-E_{mh}^{l}(j)} \tag{17}$$

4) Compute the weight W_l using

$$W_l = \left(1 - \min_{m,h} J_{mh}^{l}\right)^2 \tag{18}$$

where J_{mh}^{l} is the average of the membership function values of I^{th} user givenby:

$$J_{mh}^{l} = \frac{\sum_{q=1}^{n} M_{mh}^{l}(q)}{n}; m \neq h$$

This average must be closer to 1; hence called the unity constraint.

5) Evaluate $H_{mh}(l), h = 1, 2, \cdots, s$ using Hanman Transform in Equation (14) as

$$H_{mh}(l) = \sum_{q=1}^{n} E_{mh}^{l}(q)\mathrm{e}^{-E_{mh}^{l}(q)M_{mh}^{l}(q)} \tag{19}$$

for $m \neq h$ and $m, h = 1, 2, \cdots, s$ and $l = 1, 2, \cdots, C$

And compute K_l from:

$$K_l = \min_{m,h} H_{mh}(l) \tag{20}$$

where l stands for I^{th} user.

6) Repeat Steps 1 - 4 for all users ($l = 1, 2, \cdots, C$) and if $l = \arg\min_l \left\{\left(K_l W_l\right)\right\}$, then the test user gets identified with I^{th} user.

3.2. Algorithm-2 of Weighted Hanman Transform (WHT) Classifier

In this algorithm, Steps 1 - 3 of the above algorithm are the same as the above.

1) Compute the weight

$$w^{l}(q) = \left[1 - M_{mh}^{l}(q)\right]^2 \tag{21}$$

2) Evaluate $H_{mh}(l), h = 1, 2, \cdots, s$ using the Weighted Hanman Transform, expressed as

$$H_{mh}(l) = \sum_{q=1}^{n} w^{l}(q) E_{mh}^{l}(q)\mathrm{e}^{-E_{mh}^{l}(q)M_{mh}^{l}(q)} \tag{22}$$

for $m \neq h$ and $m, h = 1, 2, \cdots, s$ and $l = 1, 2, \cdots, C$

3) Repeat Steps 1 - 5 for all users and then find $H(l) = \min H_{mh}(l)$ for each user.

4) The I^{th} user for which $H(I)$ is minimum gives the identity of the user.

3.3. Steps for Classification

The steps to be followed for the classification are:

1) For every image, take $n \times n$, compute a find feature vector by selecting one of the features derived in Section 2.4.

2) Divide the entire feature set into the training and testing sets.

3) For every test image, apply any classifier.

4) Compute the accuracy.

4. Description of Deep Learning Neural Architectures

Various deep learning networks have emerged in the past and are still emerging because of growing interest in researchers to try them for different industrial applications. They are a popular choice for the solution of image processing problems. A few such networks used in this paper are briefly described.

4.1. Vgg-16

VGG-16 is a dense CNN introduced in 2014 by Visual Geometry Group from Oxford [16]. It was developed for ImageNet Large Scale Visual Recognition Challenge (ILSVR). It contains 16 convolution layers with only 3 × 3 convolutions and multiple filters stacked over each other. Two Fully-Connected (FC) layers with 4096 nodes and one FC with 1000 nodes are followed by a softmax classifier at the top. The average of RGB values is subtracted from images at the pre-processing stage. It has been used as one of the most prominent baseline CNN architectures in object detection and recognition problems. However, it has 138 million parameters that pose a challenge to train it from scratch. It has achieved top-5 accuracy of 90.1% on ImageNet dataset. But its memory consumption and computational cost are high.

4.2. InceptionV3

InceptionV3 was introduced in [17]. This architecture contains multiple kernel sizes (5 × 5, 3 × 3, 1 × 1) to capture information at varied scales. To reduce dimensionality, 1 × 1 convolutions are applied before going for larger kernels. InceptionV3 is a 42 layers-deep network. It is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Average pooling layer is applied after the last convolution layer which reduces the number of parameters drastically as compared to those of FC layer. This network is reported to have 5.6% top-5 error on ILSVR 2012 for classification.

4.3. ResNet-50

Kaiming He *et al.* [18] have introduced Residual Neural Network (ResNet) in the year 2015. A network with a residual block is the one where a layer feeds into its

next layer followed by another layer a few hops ahead. This is also known as skip connection and it helps overcome the degradation problem of deeper networks. It is observed that deeper networks get saturated as training proceeds and the parameters are not properly learned. These residuals skip the training of few layers thereby increasing the performance. The identity shortcuts are directly used when the input and output are of the same dimension. Each ResNet block is either two layers or three layers deep. ResNet-50 is 50 layers-deep architecture and each block has three convolution layers with the corresponding output sizes are [56 × 56, 28 × 28, 14 × 14, 7 × 7]. The three layers involve 1 × 1, 3 × 3 and 1 × 1 convolutions where 1 × 1 layers are responsible for reducing and then increasing dimensions and 3 × 3 layers for smaller input/output dimensions. It is followed by fully connected softmax. As mentioned in [18] this network achieves 3.57% on ImageNet dataset.

4.4. MobileNet

MobileNet [19] is a light-weight CNN built by Google to overcome the high memory and resource consumption problems with embedded devices. These can perform classification, detection, embeddings and segmentation like other CNNs on devices like phones which have very limited memory. The reason why MobileNet is computationally light is mainly because it uses depth-wise separable convolutions unlike other networks. Depth-wise separable convolution consists of two layers; the depth-wise convolution and the point-wise convolution. In the depth-wise convolution, one filter per input channel is used while in the normal convolution, one filter is used for all the input channels simultaneously and its computational cost is directly proportional to the spatial dimension of the feature maps and also to the number of the input and output channels while for a MobileNet it is proportional to the spatial dimension of the feature maps and the number of output channels. Thus, the cost of the computation is reduced effectively. The output of this depth-wise convolution is linearly combined using *1x1* filter in the point-wise convolution. MobileNet is usually not as accurate as other larger and resource intensive networks but it is much faster. This accuracy and resource trade-off can be further tuned by two hyper-parameters in MobileNet: width multiplier and resolution multiplier. The width multiplier is used to thin the network, while the resolution multiplier changes the input dimensions of the image thereby reducing the internal representation at every layer.

5. Databases

We have used two video databases: YouTube Faces (YTF) in [20] and UPNA Head Pose (UPNA) in [21] as these are publicly available.

5.1. YTF

This database contains 1595 subjects with varying poses, expressions, occlusions

and illuminations. The number of videos varies from 1 to 6 for all subjects. We have considered the first video of each subject. Frames (referred as images further) are extracted from videos at a rate such that the smallest set has 48 frames while the longest video contains 6070 frames. We have a total of 292,192 frames for all subjects. But we have considered only 200 subjects with 50 frames per subject. Out of 200 subjects, some have lesser than 50 frames, so a total of 9986 frames is used in all experiments.

5.2. UPNA

This database was created mainly for tracking of heads and the estimation of poses. There are10 subjects with 12 videos in each subject, 6 males and 4 females. Each video is 10 s long and contains 300 frames. In each subject, 6 videos are with guided-movement sequences and remaining 6 videos are with free-movement sequences. In the guided sequences, the user follows a specific movement, *i.e.* translation in three spatial axes and rotations comprising roll, yaw and pitch. In the free sequences, the user moves his/her head at free will while making translations and rotations along the three spatial axes. The ranges of movement are large, translations going up to more than 200 mm in any axis from the starting point, and rotations ranging up to 30˚. For experimental purpose, we have considered only one video of each subject making free movements. Out of 6 such videos per subject, one is randomly chosen. The number of frames extracted from each video is 50 to avoid time complexity. A rate of extraction was set such that 50 frames are obtained from over the entire length of each video and are not the continuous frames.

6. Results of Experimentation

The experiments are carried out on Intel core i7 processor with 2.70 GHz and 8 GB of RAM. The recognition accuracy is computed based on the correct classification of the test frames. The results are obtained by following the steps given in Section 3.3.

The performance of the information set features extracted in Section 2.4 is shown in Table 1 using different classifiers. The value of α is selected as 2.0 experimentally as the power of the information source (input) values. The algorithm is tested at different window sizes and the best results are obtained with features extracted from window of 35. Here a few, *i.e.* 10% of images are considered per person for training to reduce the computational cost. Out of 9986 images in YTF, only 989 are used for training and the remaining for the test while for UPNA out of total 500 images, 50 are used for training and 450 for testing.

Table 1 also gives the results of face recognition by Hanman Transform (HT) classifier, Constrained Hanman Transform (CHT) classifier and Weighted Hanman Transform (WHT) classifier. For some feature types, HT and CHT are giving the same results. But two feature types SG and RE show good performance on the classifiers compared. Comparative results are obtained with sigmoid (SG)

Table 1. Recognition accuracy in (%) of the features with different classifiers.

YTF						
Feature	**HT**	**CHT**	**WHT**	**LR**	**SVM**	**KNN**
EN	82.22	88.80	96	94.30	95.41	95.23
SG	**99.23**	95.71	97.78	97.70	98.18	98.11
EI	85.01	74.34	30.56	96.74	97.77	97.65
RE	99.07	96.63	**99.56**	97.30	98.09	97.99
ARE	98.93	94.55	98.89	97.03	97.73	97.54
UPNA						
Feature	**HT**	**CHT**	**WHT**	**LR**	**SVM**	**KNN**
EN	97.56	**100**	97.56	98	97.55	94.44
SG	**100**	**100**	**100**	**100**	**100**	96.44
EI	60.67	99.78	60.67	93.56	91.78	91.78
RE	**100**	92.67	**100**	**100**	89.11	87.56
ARE	94	78	95.56	99.11	60.22	95.78

features on YTF using HT and SVM with recognition accuracies of 99.23% and 98.18% respectively but the highest accuracy of 99.56% is obtained with WHT on RE features. The SG and RE features also give consistent results on UPNA with the classifiers used. The information set features outperform the deep learning (CNN) architectures' features on YTF but both information set features and CNN architectures' features demonstrate a comparable performance on UPNA. The computation required with information set features is much less than those of CNN architectures.

We have tried to cut down the computation cost by considering only 10% of the data as training set and the remaining as the testing test. To validate our approach on YTF, we have compared the results of CNNs on only 200 subjects. To reduce the computation time, we have considered 50 (few subjects have lower than 50 images) frames per subject randomly chosen.

Similarly, for UPNA there are 10 subjects with 50 frames each. 10% of data is used in training and remaining in testing.

The input tensor size is taken as 224 × 224. The input sizes, kernels, degrees and parameter values are experimented to get the best results shown in Table 2. On the classifier front Support Vector Machine (SVM) with the polynomial kernel of degree 2 is found to be the best. Another classifier called K-Nearest Neighbors (KNN) is tried for different values of K but K = 1 gives the best results.

Table 2. Recognition accuracy with various CNN architectures.

YTF			
CNN Architecture	**LR**	**SVM**	**KNN = 1**
InceptionV3	98.08	98.18	98.16
ResNet-50	52.02	52.70	92.19
Vgg-16	98.75	98.74	98.74
MobileNet	**98.84**	98.73	98.73
UPNA			
CNN Architecture	**LR**	**SVM**	**KNN = 1**
InceptionV3	98.89	97.33	98.22
ResNet-50	9.11	9.11	84.89
Vgg-16	**100**	**100**	96.67
MobileNet	**100**	**100**	89.11

As far as computational speed is concerned Vgg-16 is the slowest and MobileNet is the fastest due to their architectures. Coming to the effectiveness of features barring ResNet-50, the features from other architectures display consistent performance on both YTF and UPNA with LR, SVM and KNN classifiers. The highest score of 98.84% shown highlighted is obtained with LR on YTF. Both Vgg-16 and MobileNet have outperformed InceptionV3 and ResNet-50. Note that while extracting features using deep learning and information set-based methods, the input images are not subjected to any kind of pose and illumination correction.

As can be noticed from the above the main problem with the information set features is the choice of window size/sub-image and feature type whereas the problems with deep learning architectures include: Choice of architecture, Number of convolution and max pooling layers, activation function, and the number of filters to be used. The problems associated with the extraction of information set features can be easily fixed and the computational burden is also less; hence these features are more preferable.

7. Conclusions

An attempt has been made to make a comparative study between the deep learning features and information set features using several well-known classifiers on face videos. This study is necessitated to wean away from the blind following of deep learning methods and the related architectures for the solutions to all kinds of problems. There is a problem of choice in the ordering of the convolution and max-pooling layers in these architectures. It has been found that consideration of a large number of layers need not be accompanied with

commensurate performance. Of all the deep learning architectures, MobileNet is found to be the best followed by Vgg-16.

An alternate approach that deals with information set-based features is mainly concerned with certainty or uncertainty in the attribute or information source values, which is found using entropy functions. They provide a lot of flexibility in the generation of different types of features. In this paper, a few information set-based features have been derived followed by the formulation of Constrained Hanman Transform (CHT) and Weighted Hanman Transform (WHT) classifiers. Two information set features called Sigmoid and Renyi entropy fare extremely well on all classifiers in two datasets YTF and UPNA whereas InceptionV3, Vgg-16 and MobileNet fare extremely well on LR, SVM and KNN.

The main contributions of the paper include: 1) Promulgation of logical operations on information sets and their properties 2) Derivation of Hanman transform-based classifiers and 3) Application of both deep learning and information set based features for the video-based face recognition.

The overall performance of deep learning methods appears somewhat inferior to that of information set features with HT-based classifiers that are computationally very fast. The information set theory offers flexibility in feature extraction and classifier construction.

Our future work is concerned with extending the theory to differential entropy functions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhengm J., Ranjanm R., Chenm C., Chenm J., Castillo, C.D. and Chellappa, R. (2020) *IEEE Transactions on Biometrics, Behavior, and Identity Science*, **2**, 194-209. https://doi.org/10.1109/TBIOM.2020.2973504

[2] Wang, M. and Deng, W. (2018) Deep Face Recognition: A Survey. https://doi.org/10.1016/j.neucom.2020.10.081

[3] Inoue, H. (2018) Data Augmentation by Pairing Samples for Images Classification. https://arxiv.org/abs/1801.02929

[4] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M, *et al.* (2014) Generative Adversarial Nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Volume 2, 2672-2680. https://dl.acm.org/doi/10.5555/2969033.2969125

[5] Xiong, L., Karlekar, J., Zhao, J., Feng, J., Pranata, S. amd Shen, S. (2017) A Good Practice towards Top Performance of Face Recognition: Transferred Deep Feature Fusion. https://arxiv.org/abs/1704.00438

[6] Hanmandlu, M. and Singhal, S. (2017) Applied Soft Computing, **53**, 396-406. https://doi.org/10.1016/j.asoc.2017.01.014

[7] Hanmandlu, M. (2011) *Defence Science Journal*, **61**, 405-407. https://doi.org/10.14429/dsj.61.1192

[8] Hanmandlu, M. and Das, A. (2011) *Defence Science Journal*, **61**, 415-430. https://doi.org/10.14429/dsj.61.1177

[9] Mamta and Hanmandlu, M. (2014) *Engineering Applications of Artificial Intelligence*, **36**, 269-286. https://doi.org/10.1016/j.engappai.2014.06.028

[10] Sayeed, F. and Hanmandlu, M. (2017) *Knowledge and Information Systems*, **52**,485-507. https://doi.org/10.1007/s10115-016-1017-x

[11] Agarwal, M. and Hanmandlu, M. (2016) *IEEE Transactions on Fuzzy Systems*, **24**, 1-15. https://doi.org/10.1109/TFUZZ.2015.2417593

[12] Hanmandlu, M., Bansal, M. and Vasikarla, S. (2020) *Journal of Modern Physics*, **11**, 122-144. https://doi.org/10.4236/jmp.2020.111008

[13] Bhatia, A. and Hanmandlu, M. (2018) *Journal of Modern Physics*, **9**, 112-129. https://doi.org/10.4236/jmp.2018.92008

[14] Grover, J. and Hanmandlu, M. (2018) *Applied Intelligence*, **48**, 3394-3410. https://doi.org/10.1007/s10489-018-1154-x

[15] Grover, J. and Hanmandlu, M. (2020) *Applied Intelligence*. https://doi.org/10.1007/s10489-020-01881-3

[16] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org./abs/1409.1556v6

[17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) Rethinking the Inception Architecture for Computer Vision. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[18] He, K.M., Zhang, X.Y., Ren, S.Q. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[19] Howard, A.G., *et al.* (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861

[20] Wolf, L., Hassner, T. and Maoz, I. (2011) Face Recognition in Unconstrained Videos with Matched Background Similarity. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Colorado Springs, 20-25 June 2011, 529-534. https://doi.org/10.1109/CVPR.2011.5995566

[21] Ariz, M., Bengoechea, J.J., Villanueva, A. and Cabeza, R. (2016) *Computer Vision and Image Understanding*, **148**, 201-210. https://doi.org/10.1016/j.cviu.2015.04.009