HUMAN AGE ESTIMATION BASED ON FACE IMAGES ON CNN

E NARAYANA SWAMY[1], ANNAPURNA BAI P[2]
ASSISTANT PROFESSOR, GATES INSTITUTE OF TECHNOLOGY, GOOTY
ASSISTANT PROFESSOR, ST. MARTIN'S ENGINEERING COLLEGE, DHULAPALLY, SECUNDERABAD 500100, TELANGANA STATE, INDIA

ABSTRACT: Age estimation based on the human face remains a significant problem in computer vision and pattern recognition. In order to estimate an accurate age or age group of a facial image, most of the existing algorithms require a huge face data set attached with age labels. This imposes a constraint on the utilization of the immensely unlabeled or weakly labeled training data, e.g. the huge amount of human photos in the social networks. These images may provide no age label, but it is easily to derive the age difference for an image pair of the same person. To improve the age estimation accuracy, we propose a novel learning scheme to take advantage of these weakly labeled data via the deep Convolutional Neural Networks (CNNs). For each image pair, Kullback-Leibler divergence is employed to embed the age difference information. The entropy loss and the cross entropy loss are adaptively applied on each image to make the distribution exhibit a single peak value. The combination of these losses is designed to drive the neural network to understand the age gradually from only the age difference information. We also contribute a dataset including more than one hundred thousand face images attached with their taken dates.

1. INTRODUCTION

Face recognition is one of the biometric methods to identify individuals by features of the face. The biometric has a significant advantage over traditional authentication techniques as the biometric characteristics of the individual are unique for every person. A problem of personal verification and identification is an actively growing area of research. Face, voice, fingerprint, iris, ear, retina are the most commonly used authentication methods.

As an important biological information carrier, the human face reflects lots of properties such as identity, age, gender, expression, and emotion. With the passage of time, the facial appearance changes as human aging, which indicates human behaviour and preference. Human age can be directly inferred by distinct patterns from the facial appearance. For the same person, the photos taken at different years reveal the aging process on their faces. Age information plays an important role in human computer interaction and Artificial Intelligence systems and shares many in other face-related tasks such as face detection and recognition. Image based human age estimation has wide potential practical applications, e.g., demographic data collection for supermarkets or other public areas, age-specific human computer interfaces, age-oriented commercial advertisement, and human identification based on old ID-photos. Estimating age from images has been historically one of the most challenging problems within the field of facial analysis. With the rapid advances in computer vision and pattern recognition, computer-based age estimation on faces becomes a particularly interesting topic. However, human estimation of facial age is usually not as accurate as other kinds of facial information such as identity and gender. It is very challenging to accurately predict the age of a given facial image because human facial

aging is generally a slow and complicated process influenced by many internal and external factors.

2 EXISTING SYSTEM

The face images of 50 persons are captured by means of a digital camera (NIKON Coolpix L10). This paper proposed a novel and effective age group estimation using face features from human face images. This process involves three stages: Pre-processing, Feature Extraction, and Classification

2.1.1 *Pre-Processing:*

Face recognition is one of the biometric methods to identify individuals by features of the face. The biometric has a significant advantage over traditional authentication techniques as the biometric characteristics of the individual are unique for every person. A problem of personal verification and identification is an actively growing area of research. Face, voice, fingerprint, iris, ear, retina are the most commonly used authentication methods. Research in those areas has been conducted for more than 30 years. Experimented results are mentioned in section IV. Finally, the conclusions are in section V. Traditionally, face recognition uses for identification of documents such as land registration, passports, driver's licenses, and recognition of a human in a security area. Face images are being increasingly used as additional means of authentication in applications of high security zone. But with age progression the facial features changes and the database needs to be updated regularly which is a tedious task. So we need to address the issue of facial aging and come up with a mechanism that identifies a person in spite of aging.

2.1.2 *Feature Extraction:*

A combination of global and grid features are extracted from face images. The global features such as distance between two eye balls, eye to nose tip, eye to chin, and eye to lip is calculated in Fig.1. 2. Using four distance values, four features F1, F2, F3, and F4 is calculated as follows:

F1 = (distance from left to right eye ball) / (distance from eye to nose)

F2 = (distance from left to right eye ball) / (distance from eye to lip)

F3 = (distance from eye to nose) / (distance from eye to chin)

F4 = (distance from eye to nose) / (distance from eye to lip)

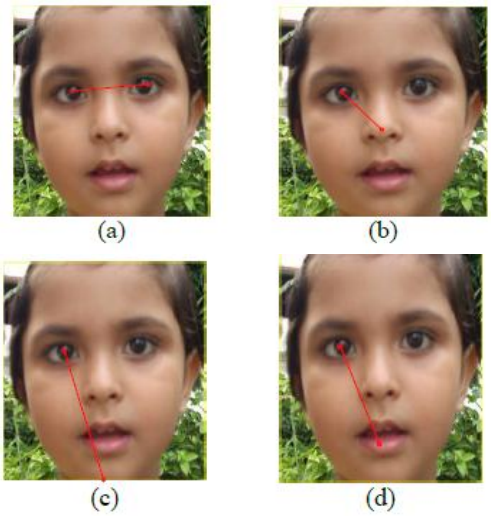


Fig 1:Face Feature Distance between (a) two eyeballs (b) eye to the nose tip (c) eye to chin (d) eye to lip

Using the Grid features of face image, feature F5 is calculated. It is entirely based on wrinkle geography in face image. The grid feature includes forehead portion, eyelid regions, upper portion of cheeks and eye corner regions as shown in Fig.3.1.3(a). To calculate feature F5, the following steps have to be followed: The color face image is converted into gray scale image. Then canny edge detection technique is applied on gray scale face image. It gives a binary face image with wrinkle edges as shown in Fig.3.1.3(b). The white pixels of the wrinkle regions in Fig. 3.1.3(b) give wrinkle information in the face image.

(a)

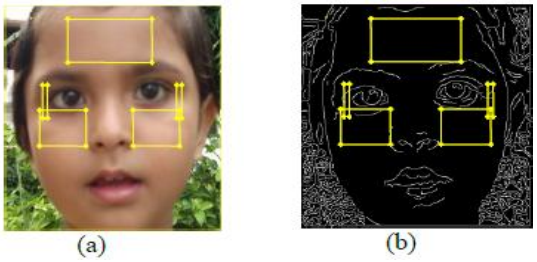


Fig 2:Face Regions (a) Grid features region of face image (b) Canny edges of face image

3.PROPOSED SYSTEM

Training the deep age difference estimator requires face images with year labels. There are numerous resources of such images on the websites such as Filckr.com where a huge number of human photos are available with taken and uploaded dates. To build our dataset, we crawled millions of photos by the query names from LFW dataset. Not only the raw images, the surrounding text which contains the related information of photos such as description and taken date is also collected. Notice that during all the pre-processing steps and experiments, we always store the image data by the query names such that the images from the same subject will stay under the same path. The pre-processing of the face dataset is shown below. It is including two flows: image processing and text processing. For the text part, we extract the taken year information from surrounding text and attach it as the related image label. All the face images are normalized to 128 _ 128. Some examples of the dataset are shown below. With the face images labeled with their taken dates, we aim to explore the age information from the difference of ages. In this work, we take advantage of age difference information to improve the age estimator. illustrates the deep architecture of the proposed approach. We first pre-train an age estimator based on the FG-NET and MORPH aging datasets via deep CNNs with multi-label loss function. For the nonage- labeled dataset, two images from the same subject are combined as a pair. Then we fine tune the whole network with the image pairs to improve the estimator.

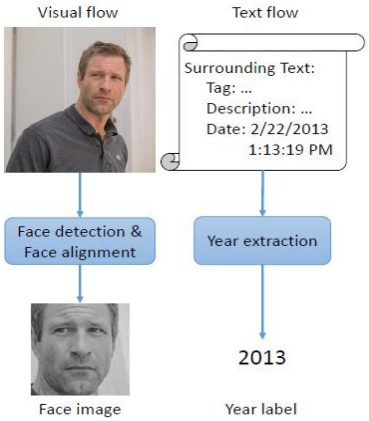


Fig.3. The age difference dataset construction

First, we train an age estimator based on the existing aging dataset. Given facial images with their ages, the age model should provide consistent estimated ages for these images. In this step, we follow the work of Geng et al. and explore the label distribution in the loss function. The advantages of label distribution, especially for age estimation task, has been demonstrated in many research works. All the existing aging datasets are labeled with given ages. Thus most algorithms treat the age estimation as a single label classification problem. However, human aging is generally a slow and smooth process in reality. The faces look quite similar at close ages. Geng proposed the typical label discrete distribution, e.g., Gaussian distribution, for the facial images. Label distribution not only can increase the number of labeled data but also tends to learn the similarity among the neighboring ages. In this paper, we use Gaussian distribution to model the label distribution of ages.

In this step, we aim to estimate the age difference between two faces. For the

images without age label, we utilize the age difference to train an age difference estimator. Given a pair of images n and m with year labels, we consider the difference of years K as the age difference. In this section, all the pair images are from the same person. Through the shared sub-network with stacked convolution layers, two images are both mapped into c-dimensional probability distributions Qn and Qm across C classes of ages. In order to explore the age information from the age difference, we carefully design three kinds of loss functions to leverage the age probability n distributions. According to the definition of softmax,Qnk = exp(fnk)=Pck=1 exp(fnk). fn is the c-dimension alintermediate feature of the output of the shared sub-network

”Convolutional neural network (CNN) is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field”. They are biologically-inspired invariant of Multilayer Perceptrons (MLP) which are designed for the purpose of minimal preprocessing. These models are widely used in image and video recognition. When CNNs are used for image recognition, they look at small portions of the input image called receptive fields with the help of multiple layers of small neuron collections which the model contains. The results we get from this collection are tiled in order for them to overlap such that a better representation of the original image is obtained; every such layer repeats this process.

This is the reason they are able if the input image is translated in any way. The outputs of neuron clusters are combined by local or global pooling layers which may be included in convolutional networks. Inspired by biological process, convolutional networks also contain various combinations of fully connected layers and convolutional layers, with point-wise nonlinearity applied at the end of or after each layer. The convolution operation is used on small regions so as to avoid the situation when if all the layers are fully connected billions of parameters will exist. Convolutional networks use shared weights in the convolutional layers i.e. for each pixel in the layer same filter (weights bank) is used which is advantageous because it reduces the required memory size and improves performance. CNNs use relatively less amount of pre-processing as compared to other image classification algorithms, meaning that the network learns the filters on its own which are traditionally manually-engineered in other algorithms. CNNs have a major advantage over others due to the lack of a dependence on prior-knowledge and the difficult to design hand-engineered features.

CNNs enforce a local connectivity pattern between neurons of adjacent layers to exploit spatially-local correlation. We have illustrated in fig.4.1 that in layer m the inputs of hidden units are from a subset of units in layer m-1, units containing spatially adjoining receptive fields.

Let us consider layer m-1 as an input retina. It can be seen in the figure that the layer m have receptive fields of width 3 in the input retina and are thus connected only to 3 adjacent neurons in the retina layer [6]. There is similar connectivity between the units in layer m+1 and the layer below. It can be said that their with respect to the input receptive field is larger where as with respect to the layer below their receptive field is 3. There is no response in the each unit to variations which are outside their receptive fields with respect to the retina thus ensuring that the strongest response to a

spatially local input pattern is produced by the learnt filter.

Every filter h_i in CNNs is duplicated across the complete visual field. The duplicated filters consists of the same parameters i.e. weights and bias that form a feature map. We can see in fig.4.2 that same feature map contains 3 hidden units. The weights of same color are shared that are constrained to be identical [6]. We can still use gradient descent to learn such shared parameters by altering the original algorithm by a very small margin. When the gradients of the shared parameters are summed, then it gives the gradient of a shared weight. We can detect the features regardless of their location in the visual field by duplicating the units. The huge reduction of the number of free parameters being learnt can lead to weight sharing increasing the learning efficiency. CNNs achieve better generalization on vision problems due to the constraints on these models.

We obtain a feature map by repeatedly applying a function across sub-regions of the entire image, mainly by convolution of the input image with a linear filter, adding a bias term Type equation here.and then applying a non-linear function. The k-th feature map can be denoted as h^k at a given layer, whose filters we can determine by the bias b^k and weights W^k , then we can obtain the feature map by the given equation: $h^{ij}_k = \tanh(W^k x)_{ij} + b_k)$.

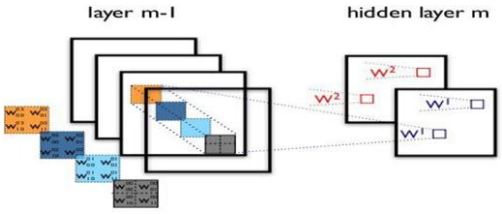


Figure 4: Convolution Layer

The fig.4 depicts 2 layers of CNN. There are 4 feature maps in layer m-1 and 2 feature maps in hidden layer m (h^0 and h^1). The pixels of layer (m-1) that lie within their 2x2 receptive field in the layer below (colored squares) are used for the computation of the pixels in the feature maps h^0 and h^1 (blue and red squares). As a result the 3D weight tensors are the weights and of and. The input feature maps is indexed by the leading dimensions, whereas the pixel coordinates is referred by the other two. layer m the weight that connects each pixel of the kth feature map with the pixel of the l-th layer at layer (m-1) and at coordinates (i,j) is denoted.

Max-pooling a form of non-linear down-sampling is an important concept of CNNs. The input image is partitioned into a group of non-overlapping rectangles and a maximum value is given for each such sub-region. We use max-pooling in vision for the following reasons-The computation of upper layers is reduced by the removal of non-maximal values. Sup-pose a max-pooling layer is cascaded with a convolutional layer. The input image can be translated by a single pixel in 8 directions. 3 out of 8 possible configurations produce exactly the same output at the convolutional layer if max-pooling is done over a 2x2 region. This jumps to 5/8 for max-pooling over a 3x3 region. A form of translation invariance is provided by this. The dimensionality of intermediate representations is reduced by max-pooling because it provides additional robustness to position.

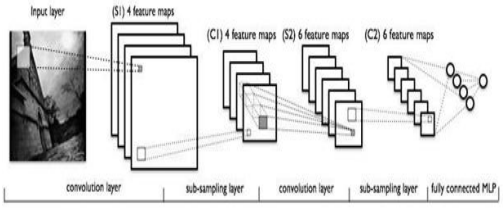


Figure 5: Full LeNet Model

The LeNet family of models have sparse, convolutional layers and max-pooling concepts as its core. The exact details of the model shown in fig.4.4 will vary a lot, it shows how a LeNet model will look like. The alternating convolution and max-pooling layers compose the lower-layers of the model. ”The upper-layers however are fully-connected and correspond to a traditional Multi-layer Perceptron which is a combination of hidden layer and logistic regression. The input to the first fully-connected layer is the set of all features maps at the layer below.

The dataset I chose for this thesis is from the SUN database. The major reason for choosing this dataset was that the images in it were pre-annotated and had annotations as XML files for each image. The SUN database is huge so I had to choose a small subset of it for this study. In this study I am trying to classify images based on 8 classes namely: water, car, mountain, ground, tree, building, snow, sky and unknown which contains all the rest of the classes. I chose only those sets of images which I felt were more relevant to these classes. I collected a database of 3000 images from 41 categories. Each image has its annotations in an XML file. I randomly divided the dataset into 80% training set and 20% testing. There are 1900 training images, 600 testing images and 500 validation images. The training set was further divided into 80% training set and 20% validation set. The major drawback of this dataset is that the images are annotated by humans and the annotations are not perfect thus it may have some effect on the results. I try to handle this problem by getting as many synonyms as I can for each class label. A few examples of the synonyms are lake, lake water, sea water, river water, wave, ripple, river, sea, river water among others which all belong to the class label water. I mapped these synonyms to their respective class labels which are being used. Not all images in every categories were annotated. I filtered out the annotated images from the dataset and used only them for this study. Fig.4.5 shows an example of an image from the dataset and its annotation file where it can be seen how a river is annotated by the user.

A little pre-processing was required on the dataset before it could be trained because of the way the code for CNN training was written. The images were converted to grayscale and resized to 28x28 pixels. I used the annotation files to get a flag for each class present or absent from the image and using the flags I compressed the dataset into a 1D array which contains the image dimensions and binary values for each class where 1 states that class is present and 0 states the class is absent. The compressed data is then trained by the neural

SYSTEM DESIGN:

The fig.4.6 illustrates how the system of retrieval works for this study. The query image is pre-processed and is evaluated with the trained neural network and the regions are clas-sified. It is then matched against the annotation index with images on which the neural network was trained. All the images in the dataset which are similar to the query image are returned to the user based on the number of images required by him. In other words, top N images similar to the query image are retrieved. This section briefly explains the major components of the system design

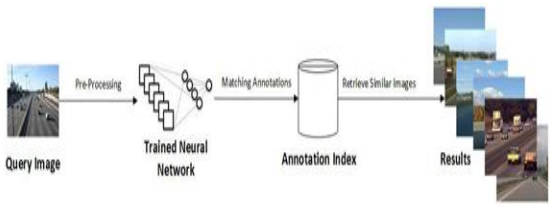
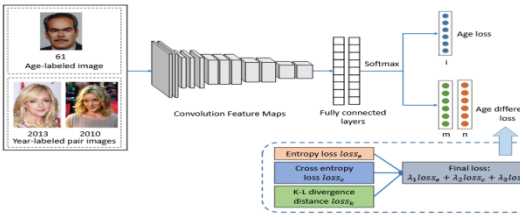


Figure 4.6: System Design.



4.7:System Architecture

Since the output of the network is the probability distribution across a possible age range, each entry indicates the probability of the age class. Given an age probability vector, the array should have a single peak, rather than be uniformly distributed. We choose the entropy loss to satisfy this requirement. Because the entropy loss will be 0 only if one entry is 1 and all others are 0. If the probabilities are uniform values, the loss will be largest. The entropy loss for the image n is defined as

$$loss_e = -\sum_{k=1}^{c} Q_{nk} \log(Q_{nk})$$

(1)

Before deriving the backward function, the gradient of Qnk with respect to fnk is

$$\frac{\partial Q_{nk}}{\partial f_{np}} = Q_{nk}\left(\delta(k=p) - Q_{np}\right)$$

(2)

The notation _(k = p) is 1 if fk = p; otherwise 0. This equation is formulated according to the definition of the soft max function. To optimize the network parameters, the gradient of loss with respect to function is

$$\frac{\partial loss_e}{\partial f_{np}} = \frac{\partial loss_e}{\partial Q_{nk}} \cdot \frac{\partial Q_{nk}}{\partial f_{np}}$$

$$= Q_{nk}\left(\delta(k=p) - Q_{np}\right) \cdot -\sum_{k=1}^{c}(\log(Q_{nk}) + 1)$$

$$= \sum_{k=1}^{c} Q_{nk}\left(\delta(k=p) - Q_{np}\right)\log(Q_{nk}) + Q_{nk}\left(\delta(k=p) - Q_{np}\right)$$

$$= Q_{np}\log(Q_{np}) - Q_{np}\sum_{k=1}^{c} Q_{nk}\log(Q_{nk})$$

(3)

If the age difference between a pair of face images n and m is K years, assuming the image n is K years younger than the image m, then the age of image n should be no more than cK years old and the age of image m should be older than K years old. According to this, we can infer that the probability values from C…..K to c elements of image n should be zero and the same for image m from 0 to K elements. Take the image n for example. We split the output of soft max layer into two parts and add up the values of elements from 0 to C …K as Q1n while the summation of remains isQ2n. This is equivalent to a binary classifier. Then we set a binary vector b = (1; 0) and implement the cross entropy loss to measure the distance between the (Q1n;Q2n) and the binary vector b. The cross entropy loss for image n is defined as

$$loss_c = -\sum_{i=1}^{2} b_i \log\left(Q_n^i\right) = -\log(Q_n^1)$$

(4)

Here Q1n =PcK k=0 Qnk. For the back propagation, the gradient of loss with respect to fnk is

$$\frac{\partial loss_c}{\partial f_{np}} = \frac{\partial loss_e}{\partial Q_n^1} \cdot \frac{\partial Q_n^1}{\partial f_{np}}$$

$$= -\frac{1}{\sum_{k=1}^{c-K} Q_{nk}}\left(\sum_{k=1}^{c-K} Q_{nk}\left(\delta(k=p) - Q_{np}\right)\right)$$

$$= Q_{np} - \frac{Q_{nk}\,\delta(k=1,\ldots..,c-K)}{\sum_{k=1}^{c-K} Q_{nk}}$$

(5)

where (k = 1 _ _ _ C… K) is 1 if the k is larger than 1 and smaller or equal to C…. K and is 0 otherwise. The output of image m is processed in

the same way into (Q1m;Q2m) and compared with b0 = (0; 1).

Given a pair of images with age difference K of the same person, the age probability distributions should be approximate after a translation of all entries with K steps. In this step, we design a translation Kullback-Leibler (K-L) divergence loss function to quantify the dissimilarity between the distributions of image n and the translated distribution of image m. We expect and the K-L divergences distance between these two probabilities is defined as

$$\mathrm{KL}(Q_n, Q_m^{'}) = \sum_{k=1}^{c} Q_{nk}\, log\left(\frac{Q_{nk}}{Q_{m\,(k+K)}}\right) \quad (6)$$

Since K-L distance is asymmetric, we make it as symmetric as

$$loss_k = \sum_k Q_{nk} \log\left(\frac{Q_{nk}}{Q_{m\,(k+K)}}\right) + Q_{m\,(k+K)} \log\left(\frac{Q_{m\,(k+K)}}{Q_{nk}}\right) \quad (7)$$

and for the image m the K-L divergence loss is

$$loss_k = \sum_k Q_{n\,(k-K)} \log\left(\frac{Q_{n\,(k-K)}}{Q_{mk}}\right) + Q_{mk} \log\left(\frac{Q_{mk}}{Q_{n\,(k-K)}}\right) \quad (8)$$

Here the Qn(K….K) is the translated probability distribution of image n.

The gradient for backward for the image n is

$$\frac{\partial loss_k}{\partial f_{np}} = \frac{\partial loss_k}{\partial Q_{nk}} \cdot \frac{\partial Q_{nk}}{\partial f_{np}}$$

$$= \sum_k Q_{nk}\left(\delta(k=p) - Q_{np}\right) \log\left(\frac{Q_{nk}}{Q_{m\,(k+K}}\right) + Q_{nk}\left(\delta(k=p) - Q_{np}\right) - \frac{Q_{m\,(k+K)}}{Q_{nk}} Q_{nk}\left(\delta(k=p) - Q_{np}\right)$$

$$= Q_{np} \log\left(\frac{Q_{np}}{Q_{m\,(k+K)}}\right) - Q_{np} \sum_k Q_{nk} \log\left(\frac{Q_{nk}}{Q_{m\,(k+K)}}\right) + Q_{np} - Q_{m\,(p+K)} \quad (9)$$

Finally, the overall loss of the whole age difference estimation network is

$$min\ \varphi = \min(\lambda_1 loss_e + \lambda_2 loss_c + \lambda_3 loss_k) \quad (10)$$

Where λ_1 , λ_2 & λ_3 are terms of tradeoff between the errors.

I have discussed in section 4.3 how I trained the neural network. The result that is returned after training is a train model which is a The no function. After the query image is converted to grayscale and is resized it is evaluated with the train model. Based on the training results the regions of the query image are classified according to the class labels. This information is stored and is used for matching against the annotation index.

6.1 RESULT SCREENSHOTS:

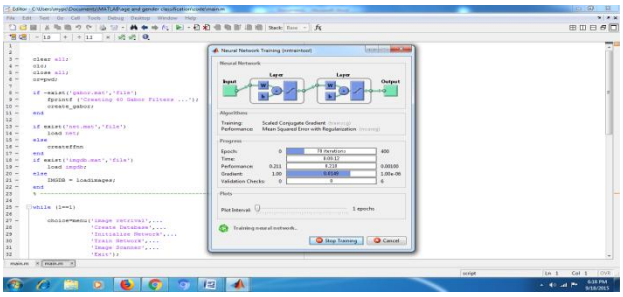


Fig: 6.11 Open the Network

After selecting train network we will get a new window called Neural Network Training(Initialize tool) and the operations are performed automatically as shown in above Fig: 6.11.

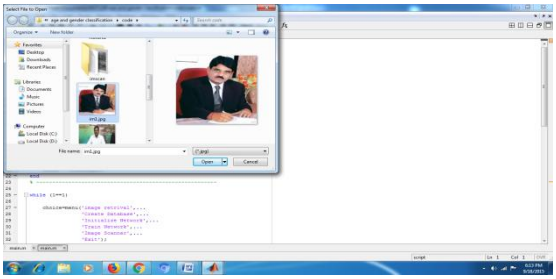


Fig: 6.12 Image Scanner

we will get back to Age Detection window and select Image Scanner, after selecting the option we will get all the images from the images we will select one image as input as shown in above Fig: 6.12.

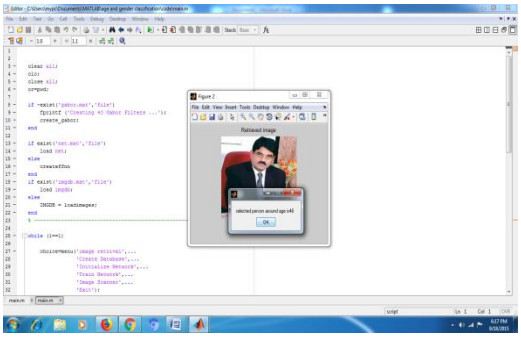


Fig: 6.15 Final output

Age detector detects the image which we gave as input at the starting as retrieved image and perform the operations on the retrieved image and finally displays the age of the input image as shown in above Fig: 6.15. where you can see a dialogue box where it is showing as selected person around age.

CONCLUSION

we mainly investigate the problem of age estimation without age label and propose an approach to estimate the age of a human face with the assistance of age difference information. Given a pair of face images taken at different years of the same subjects, we exploit the age information from the images of age difference via the deep Convolutional Neural Networks (CNNs). First, we build a deep age estimator based on the standard aging datasets. A symmetric Kullback-Leibler divergence loss function is placed at the top layer of CNNs. We utilize label distribution to design the loss function. We design three kinds of loss functions on the top of the softmax layer to learn the representation of age difference. Experimental results show the advantages of the proposed age difference learning system and the state-of-the art performance is gained.

FUTURE SCOPE:

In the future work, we aim to explore more biological features of people such as appearance, hair style, height, pose, emotions and gait.

REFERENCES

[1]A Midori Albert, Karl Ricanek, and Eric Patterson. A review of the literature on the aging adult skull and face: Implications for forensic science research and applications. Forensic Science International,172(1):1–9, 2007.

[2] Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(10):2401–2412, 2013.

[3]Jiwen Lu, Venice Erin Liong, and Jie Zhou. Cost-sensitive local binary feature learning for facial age estimation. Image Processing, IEEE Transactions on, 24(12):5356–5368, 2015.

[4]Ivan Huerta, Carles Fernandez,´ Carlos Segura, Javier Hernando, and Andrea Prati. A deep analysis on age estimation. Pattern Recognition Letters, 68:239–249, 2015.

[5]Kuang-Yu Chang and Chu-Song Chen. A learning framework for age rank estimation based on face images with scattering transform. Image Processing, IEEE Transactions on, 24(3):785–798, 2015.

[6]Hamdi Dibeklioglu, Fares Alnajar, Albert Ali Salah, and Theo Gevers. Combining facial dynamics with appearance for age estimation. Image Processing, IEEE Transactions on, 24(6):1928–1943, 2015.

[7]Yun Fu, Guodong Guo, and Thomas S Huang. Age synthesis and esti- mation via faces: A survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(11):1955–1976,2010.

[8]Guodong Guo, Guowang Mu, Yun Fu, and Thomas S Huang. Human age estimation using bio-inspired features. In Computer Vision and Pattern Recognition, IEEE Conference on, pages 112–119, 2009.

[9]Guodong Guo, Yun Fu, Thomas S Huang, and Charles R Dyer. Locally adjusted robust regression for human age estimation. In Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision, pages 1–6, 2008.

[10]Shuicheng Yan, Huan Wang, Xiaoou Tang, and Thomas S Huang. Learning auto-structured regressor from uncertain nonnegative labels. In Computer Vision, IEEE 11th International Conference on, pages 1–8, 2007.

[11]Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3476– 3483, 2013.