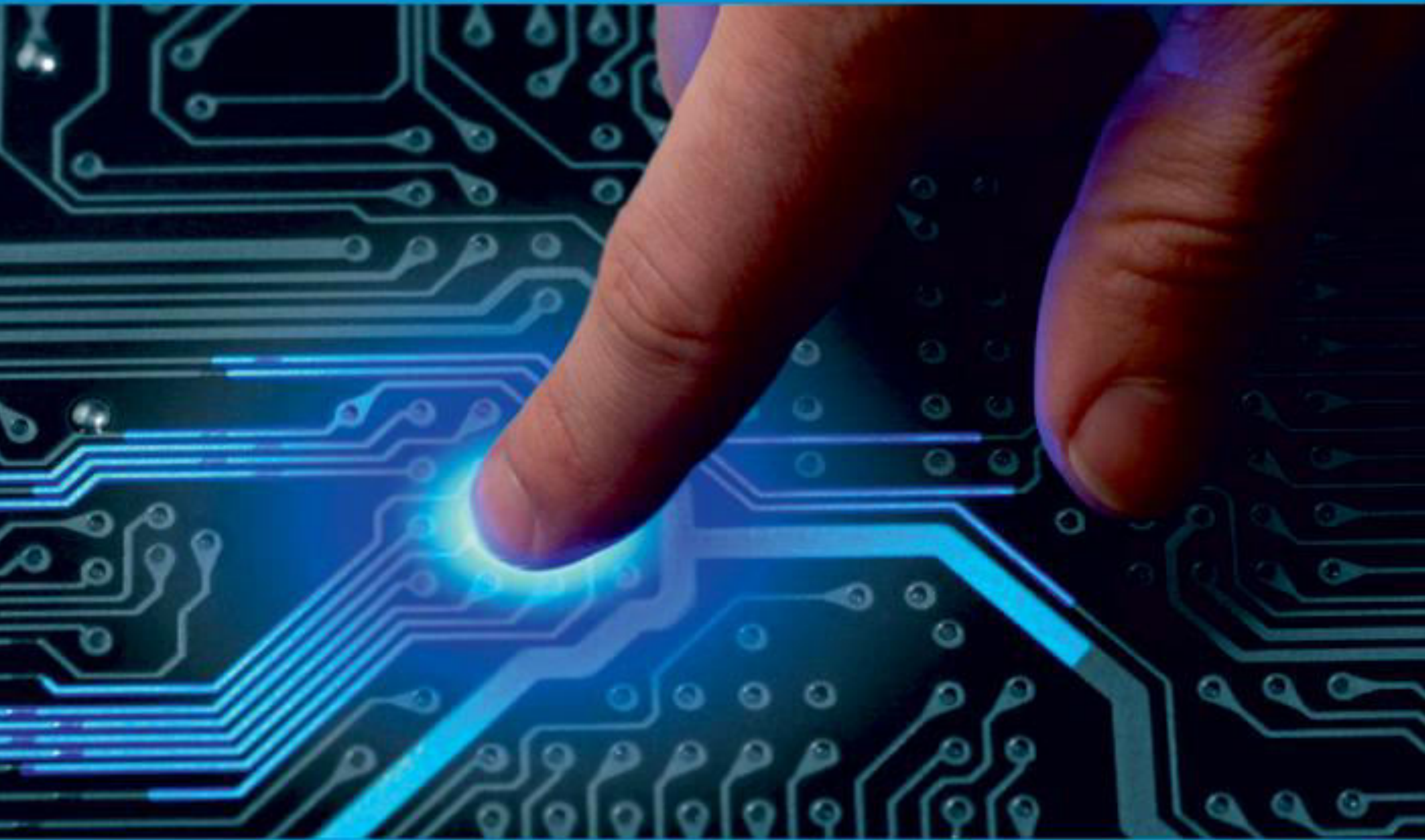
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 8, August 2024| DOI: 10.15680/IJIRCCE.2024.1208009

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Optimizing Image Extraction: WLS Method Enhances Processing and System Quality

Surendra Kumar, Dr. Jitendra Rai, Dr. K. P. Jayant

Research Scholar, Department of Computer Science and Engineering, Sunrise University, Alwar, Rajasthan, India

Department of Computer Science and Engineering, Sunrise University, Alwar, Rajasthan, India

Department of Computer Science and Engineering, Sunrise University, Alwar, Rajasthan, India

ABSTRACT: This study demonstrates how choosing the right extraction technique impacts processing times and overall system quality. Among the methods examined, the Weighted Least Squares (WLS) approach proves to be the most effective. The findings underscore the need for continued research and development in image extraction and fusion, paving the way for more advanced, effective, and adaptable methods to meet growing demands. This review article explores recent advances in shared weight networks within artificial neural networks and image processing, focusing primarily on the latest research.

KEYWORDS: - Shared Weight Networks, Artificial Neural Networks, Image Processing.

I.INTRODUCTION

1.1 Introduction

In the field of artificial intelligence and computer vision, the ability to interpret and interpret visual data is among countless applications, from facial recognition to autonomous vehicles. The key to this skill is the feature extraction process, which involves identifying and distinguishing relevant patterns, edges, and structures within images. In recent years, the intersection of advanced artificial neural networks and image processing has revolutionized our ability to extract and use these important features effectively [1]. One important approach that has emerged as a foundation in this field is the use of shared weight networks within artificial neural networks. This approach not only improves the efficiency of the Background feature but also enables significant improvements in image recognition, object detection, and scene understanding [2]. This study discusses the exciting world of weight-shared networks and their important role in image processing feature extraction. We will explore the fundamental principles behind distributed weight structures, their application to convolutional neural networks (CNNs), and the remarkable steps that have enabled them to create consistent representations of the interpretation and sequence of visual data [3]. In addition, we will explore real-world use cases, discuss the implications of weight sharing for computational efficiency, and touch on the exciting frontiers of transfer learning and custom architecture design. Feature extraction using shared weight networks in artificial neural networks is a technique commonly used in image processing and computer vision tasks. This approach involves designing neural network architectures that share certain weights or parameters across layers or units [4]. This distributed weight structure is particularly useful for tasks where translational flexibility or learning of spatial categories is important, such as image recognition and object detection. Following is an overview of how shared weight networks are used in artificial neural networks for feature extraction in image processing:

Convolutional Neural Networks (CNNs): CNNs are a class of neural networks that have become the standard for image processing tasks. They use shared weight networks in the form of convolutional layers. These layers contain adaptive filters or kernels that slide over the input image to extract local features. The weights in these filters are distributed to different areas in the image, allowing the network to capture spatial patterns and features [5].

Weight Sharing in Transform Layers: In transform layers, weight sharing means that the same set of filter weights is applied to different receptive fields in the input image. This allows the network to detect the same features (eg, edges, textures) in different locations, making it consistent [6]. Weight sharing reduces the number of parameters in the network, making it computationally efficient.

© 2024 IJIRCCE | Volume 12, Issue 8, August 2024| DOI: 10.15680/IJIRCCE.2024.1208009

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Feature Hierarchies: CNNs often have multiple layers stacked on top of each other. These layers gradually learn hierarchical features, starting with simple ones such as edges and gradually progressing to more complex features such as object parts and whole objects. Weight sharing allows each layer to build on the features learned by the previous layer, creating a sequential representation of the input image [7].

Blending Layers: In addition to blending layers, blending layers are often used to downsample feature maps and reduce spatial dimensions. Aggregation layers also use weight sharing, typically using maximum aggregation or average aggregation functions. This further improves the network's ability to capture consistent features [8].

Transfer Learning: Pre-trained CNNs, such as those trained on large datasets such as ImageNet, can be used as feature generators for various image processing tasks. By setting the weights of the convolutional layers and training only the fully connected layers over the network, you can optimize the features learned by the pre-trained model for tasks such as image segmentation, object detection, or segmentation.

Custom Architectures: In some cases, researchers design custom architectures with shared weight structures that suit specific image processing tasks. For example, Siamese networks use session sharing do to learn the similarity between pairs of images, while U-Net architectures use weight sharing for image classification tasks.

II.RESEARCH METHODOLOGY

2.1 Introduction

When it comes to MSDF, it is described as the process of integrating data from various sources for the most accurate and comprehensive unified data on an object, activity, or event to be produced. Kalman filter can play a great role in several traditional engineering field as well as emerging technological devices. A high-level representation of multi-Feature y DF is shown in Figure presented below. Specifically, three alternative Extraction techniques are explored and implemented in a tracking system in this work. There is a computation of performance measures, and the findings gained are presented in a descriptive manner.

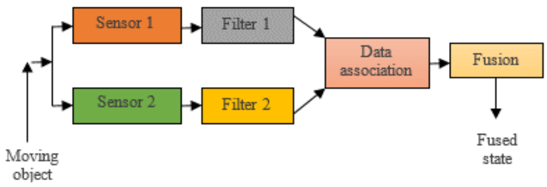


Fig 2. 1: Concept of Multi extraction Data Extraction (DF)
Source: Created by Researcher

In last decade, the rapid change has been notices in technology specially in camera and photography. The use of extraction in camera has been largely investigated by various scientific community around the world. It is possible for multi-feature extraction to take place at the level of representation of the signal as well as at the levels of image, feature, and symbol representation, among other levels of representation. Fusing signals at the signal-level refers to the direct mixing of various signals in order to generate a signal that is essentially identical to the source signals in terms of its fundamental structure. Specifically, at the data-level, Extraction refers to the direct integration of numerous data points in order to generate a signal that has a fundamental format that is identical to that of the source data. When creating a FIM, image-level Extraction (also called picture Extraction) determines each pixel by comparing it to a collection of pixels in each source image, resulting in one FIM with one pixel for each source image. Image-level Extraction is a technique for merging two or more source images into one FIM. Many elements of image-level Extraction, also known as picture Extraction, are like those of signal-level Extraction, because an image may be conceived of as a two-dimensional signal in certain cases. We make this difference because we are primarily concerned with picture Extraction in this context.

The merging of edge maps is one of the most common types of feature-level Extraction that is used. Symbol-level Extraction is a way to combine information from a lot of different extraction and make it work well at the highest

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

level of abstraction possible. Processed information or a symbolic process may be used to come up with them. They may also be based on information not given at the time of the Extraction. Decision Extraction (also known as distributed detection) is a kind of symbol-level Extraction that is often used.

III.RESULT

CBF Technique

Local, non-iterative, and non-iterative filtering is a technique that combines low-pass filtering with an edge-stopping function that is non-iterative. A kilogram filter will be added to the edge exercise if the variation in thickness between pixels is quite significant between them.

It is a local, non-linear, and non-iterative technique. Because the weights of the filter consider both grey level similarities and geometric proximity of the nearby pixels, the weights of the filter are dependent not only on Euclidian distance-ED but also on distance in gray/coulor space. The filter's benefit is that it smooths the picture while keeping edges by utilising surrounding pixels in the process. The following is the mathematical formula for calculating the BF output at a pixel point P in an image A:

$$A_F(p) = \frac{1}{W}\sum\nolimits_{q\in s} G_{\sigma_S}(\parallel p-q \parallel) G_{\sigma_r}(IA(p)-A(q)I)A(q) \qquad (1)$$

$Where G_{\sigma_S}(\parallel p-q \parallel) = e^{\frac{\parallel p-q\parallel^2}{2\sigma_r^2}}$ is a geometric closeness function,

$G_{\sigma_r}(IA(p)-A(q)I) = e^{\frac{IA(p)-A(q)I^2}{2\sigma_r^2}}$ is a gray level likeness/ edge-stopping occupation,

$W = \sum_{q\in S} G_{\sigma_S}(\parallel p-q \parallel)\, G_{\sigma_r}(IA(p)-A(q)I)$ is a standardization constant,

$\parallel p-q \parallel$ is the Euclidean distance between p and q.

And S is a spatial neighborhood of p.

Since $\sigma_S$and σ_r control the behaviour of BF, the dependency of ${}^{\sigma_r}/_{\sigma_S}$ value and derivative of the input single on the behaviours of the BF are analysed. The best σ_S worth is
It was selected to required degree of LPF, and it blurs extra for big σ_S, because it integrated information from more distant picture locations, which causes it to blur more for large σ_S,. Additionally, if a picture is scaled up or down, the value of σ_S, must be modified to get the same effects as the original image. In general, it seems that a suitable range for the σ_S, value is around (1.5-2.1); on the other hand, the ideal σ_r value will depend on how much edge is desired to be kept. If the picture is magnified or attenuated in any way, the value of σ_r must be modified properly in order to get the desired outcome.
CBF uses both grey level and regular similarity of neighboring pixels in picture A to form the filter kernel of image B, which is filtered. A pixel position p in image B is used to compute the CBF. [31]:

$$B_{CBF}(p) = \frac{1}{W}\sum\nolimits_{q\in S} G_{\sigma_S}(\parallel p-q \parallel) G_{\sigma_r}(IA(p)-A(q)I)B(q) \qquad (2)$$

$where, G_{\sigma_r}(IA(p)-A(q)I) = e^{\frac{IA(p)-A(q)^2}{2\sigma_r^2}}$ is a gray level edge index

$and\ W = \sum_{q\in S} G_{\sigma_S}(\parallel p-q \parallel) G_{\sigma_r}(IA(p)-A(q))$ is a normalization constant.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Unfocused areas in image A focused in image B and the application of CBF blur the focused region more than that of the blurred area in image B. (Zhou & Wang, 2014) This is since the picture has a blurred region.

Fig 3.1: The above images presented the CBF technique on the images (a) to (f) with each space of blurring area visualizing in above images.

Consequently, the filter kernel is near to Gaussian in that region, making it seem blurry. Utilizing a weighted average, it is possible to estimate the weights for I-F by using the focused region details captured in detail image B_D. Image B information is not present in image A in multi-extraction pictures, hence applying CBF on image B will obscure image B information. Since the information in that area is missing, each grey level in that region has the same value making the kernel Gaussian. Simulated multi-focus lady source pictures are shown in Fig. 4.2 with their corresponding CBF output and detail images. According to Figs. 4.2 (c) and (d), CBF has blurred and preserved the focused region while capturing its details in the detail photos. A new method for determining weights has been developed using these detailed photos.

Conv_SR (CSR) Technique

A distinctive feature of CSR is that it is dedicated to increasing the quality of the Extraction , thus boosting image coverage in its entirety. Before anything further, two multimodal focus pictures are created in order to produce wavelet change as Stationary Wavelet Transform (SWT). Stationary Wavelet Transform supplies us with four sub bands, which are denoted by the letters LL, LH, HL, and HH. The cut result is obtained by the LL band via the application of CSR. The final output picture is obtained via the usage of a separate SWT.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

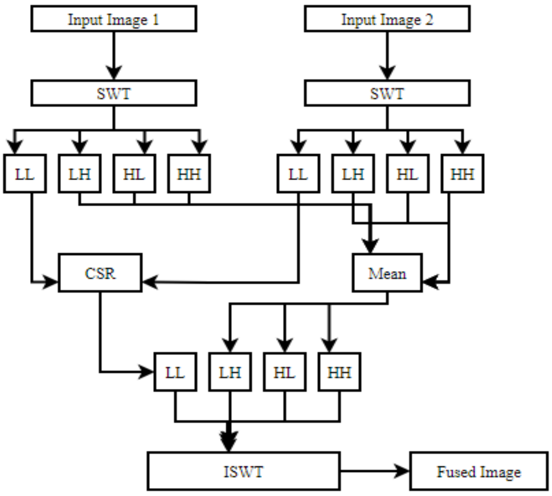


Fig3.2: Method of Conv_SR (CSR) in general

Source: Created by Researcher

WLS Method

The technique demonstrates the data completeness may be accomplished by using various infrared (IR) properties and optical qualities of the picture, with or without the usage of detail from the original image.

Simulative Investigation

The simulative study of picture Extraction has been carried out using the MATLAB programming language. The underlying process by which the Extraction of picture 1 and picture 2 is accomplished requires to complete with some execution time. Table 1 contains the results of this temporal analysis, which has been noted. This period reflects the rapidity with which the method fuses the materials. [MATLAB DB] was used to gather the source infrared and visible pictures, which were used in our simulation.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

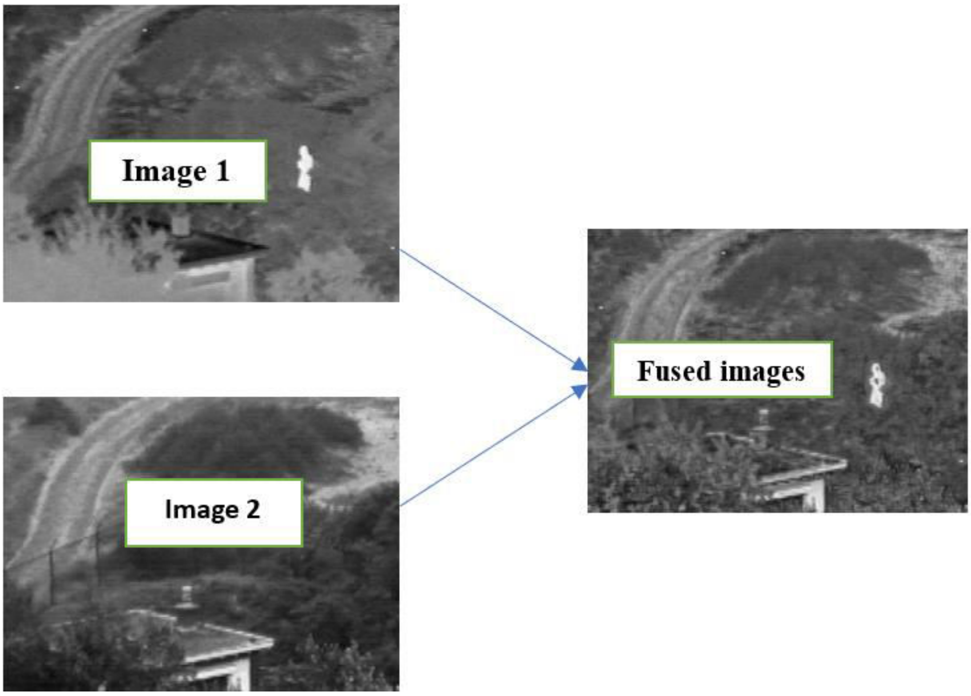


Fig 3.3: Extraction Process of Images of two different way

Source: Created by Researcher

Table 3. 1 Estimation Extraction process time obtained from each technique

Sr. No.	Number of Pixels (Image1)	Number of Pixels (Image2)	CBF	Conv-SR	WLS
1	45	46	44.62	109.77	4.22
2	124	94	116.12	543.99	6.51
3	92	64	100.42	436.90	5.41
4	93	142	135.04	429.86	6.41
5	53	44	41.29	102.63	2.21
6	168	217	199.90	634.93	10.01
7	58	70	94.74	412.84	4.01
8	38	55	38.28	132.18	1.31
9	166	266	212.39	738.03	11.20
10	166	248	216.61	599.86	9.54
11	34	40	42.90	179.00	1.49
12	137	127	133.60	474.45	6.10
13	127	147	134.58	449.77	5.94
14	147	229	345.36	655.37	9.44
15	175	234	303.83	638.49	9.52
16	111	129	178.32	397.04	6.16
17	126	102	181.09	393.00	5.91
18	135	125	189.71	365.28	7.83

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

19	138	153	168.92	438.03	5.80
20	170	189	278.04	635.31	11.45
21	140	246	259.66	669.14	9.57
Estimated Average Time			162.64	449.33	6.67

Source: Extracted data from MATLAB Simulation

Estimation of Images from Python

In dimensionality reduction, also known as "feature extraction," the raw data is broken down into smaller, more manageable chunks by a process of grouping and partitioning. So, it'll be less of a hassle when you want to really process. One crucial aspect of these massive datasets is the sheer number of variables they include. Large amounts of computational power are needed to handle these variables. This is where feature extraction comes in; by choosing and merging variables into features, it helps to obtain the best feature out of those massive data sets. These characteristics may accurately and creatively depict the real data set, while yet being simple to handle. In this part we performed it by python simulation.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

#Visualization
import matplotlib.pyplot as plt

#Machine Learning
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn import metrics
import os
```

```
print("There are {} images in the dataset".format(len(data)))
print("There are {} unique targets in the dataset".format(len(np.unique(target))))
print("Size of each image is {}x{}".format(data.shape[1],data.shape[2]))
print("Pixel values were scaled to [0,1] interval. e.g:{}".format(data[0][0,:4]))
```

```
There are 400 images in the dataset
There are 40 unique targets in the dataset
Size of each image is 64x64
Pixel values were scaled to [0,1] interval. e.g:[0.30991736 0.3677686  0.41735536 0.44214877]
```

```
print("unique target number:",np.unique(target))
```

```
unique target number: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
def show_40_distinct_people(images, unique_ids):
    #Creating 4X10 subplots in  18x9 figure size
    fig, axarr=plt.subplots(nrows=4, ncols=10, figsize=(18, 9))
    #For easy iteration flattened 4X10 subplots matrix to 40 array
    axarr=axarr.flatten()

    #iterating over user ids
    for unique_id in unique_ids:
        image_index=unique_id*10
        axarr[unique_id].imshow(images[image_index], cmap='gray')
        axarr[unique_id].set_xticks([])
        axarr[unique_id].set_yticks([])
        axarr[unique_id].set_title("face id:{}".format(unique_id))
    plt.suptitle("There are 40 distinct people in the dataset")
```

Fig3.5 : *Feature extraction*

© 2024 IJIRCCE | Volume 12, Issue 8, August 2024| DOI: 10.15680/IJIRCCE.2024.1208009

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
def show_10_faces_of_n_subject(images, subject_ids):
    cols=10# each subject has 10 distinct face images
    rows=(len(subject_ids)*10)/cols #
    rows=int(rows)

    fig, axarr=plt.subplots(nrows=rows, ncols=cols, figsize=(18,9))
    #axarr=axarr.flatten()

    for i, subject_id in enumerate(subject_ids):
        for j in range(cols):
            image_index=subject_id*10 + j
            axarr[i,j].imshow(images[image_index], cmap="gray")
            axarr[i,j].set_xticks([])
            axarr[i,j].set_yticks([])
            axarr[i,j].set_title("face id:{}".format(subject_id))
```


```
#We reshape images for machine learnig  model
X=data.reshape((data.shape[0],data.shape[1]*data.shape[2]))
print("X shape:",X.shape)
```

```
X shape: (400, 4096)
```

```
X_train, X_test, y_train, y_test=train_test_split(X, target, test_size=0.3, stratify=target, random_state=0)
print("X_train shape:",X_train.shape)
print("y_train shape:{}".format(y_train.shape))
```

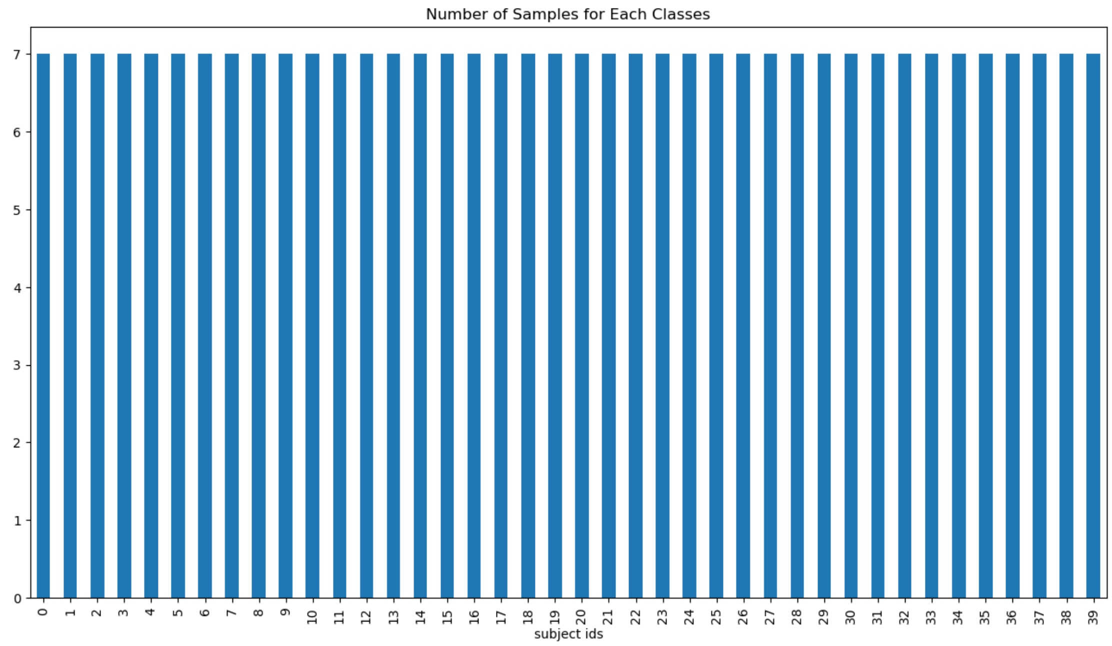
```
X_train shape: (280, 4096)
y_train shape:(280,)
```

```
y_frame=pd.DataFrame()
y_frame['subject ids']=y_train
y_frame.groupby(['subject ids']).size().plot.bar(figsize=(15,8),title="Number of Samples for Each Classes")
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



© 2024 IJIRCCE | Volume 12, Issue 8, August 2024| DOI: 10.15680/IJIRCCE.2024.1208009

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

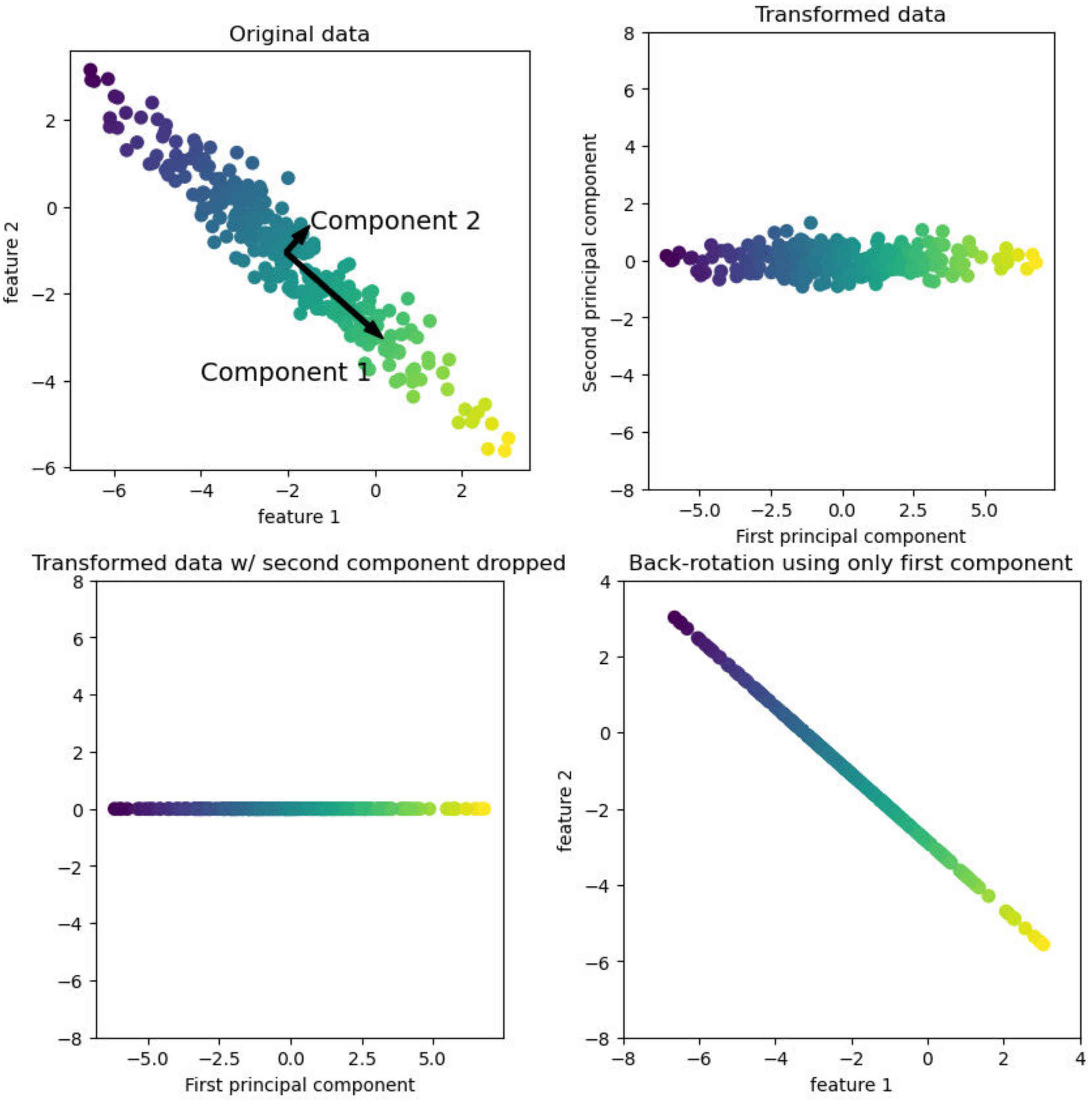


Fig 3.6: Principal component of featured face

IV.CONCLUSION

The simulation time measurement of the image extraction process is an important factor to improve system performance and efficiency. This study showed the significant impact of choosing the right extraction method on processing times and overall system quality. The WLS method, in particular, stands out as the most efficient among the tested methods. The findings emphasize the importance of continued research and development in the field of image extraction and fusion, paving the way for more advanced, efficient, and versatile techniques that can meet the growing needs of various applications. The integration of deep learning methods and the optimization of existing algorithms will be important in achieving these goals, ultimately contributing to the development of technology and its use in various fields.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

1. Li, W., Yang, C., Peng, Y., & Du, J. (2022). A pseudo-siamese deep convolutional neural network for spatiotemporal satellite image Extraction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 1205-1220.
2. Dong, Y., Liu, Q., Du, B., & Zhang, L. (2022). Weighted feature Extraction of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Transactions on Image Processing*, *31*, 1559-1572.
3. Chen, S., Yu, J., & Wang, S. (2022). One-dimensional convolutional neural network-based active feature Extraction for fault detection and diagnosis of industrial processes and its understanding via visualization. *ISA transactions*, *122*, 424-443.
4. Ranjbarzadeh, R., Tataei Sarshar, N., Jafarzadeh Ghoushchi, S., Saleh Esfahani, M., Parhizkar, M., Pourasad, Y., ... & Bendechache, M. (2022). MRFE-CNN: Multi-route feature Extraction model for breast tumor segmentation in Mammograms using a convolutional neural network. *Annals of Operations Research*, 1-22.
5. Siar, M., & Teshnehlab, M. (2022). A combination of feature Extraction methods and deep learning for brain tumour classification. *IET Image Processing*, *16*(2), 416-441.
6. Anitha, K., & Srinivasan, S. (2022). Feature Extraction and Classification of Plant Leaf Diseases Using Deep Learning Techniques. *Computers, Materials & Continua*, *73*(1).
7. Zhang, C., Feng, Y., Hu, L., Tapete, D., Pan, L., Liang, Z., ... & Yue, P. (2022). A domain adaptation neural network for change detection with heterogeneous optical and SAR remote sensing images. *International Journal of Applied Earth Observation and Geoinformation*, *109*, 102769.
8. Zhang, Z., & Wang, M. (2022). Convolutional neural network with convolutional block attention module for finger vein recognition. *arXiv preprint arXiv:2202.06673*.