Journal of Theoretical and Applied Information Technology
31st August 2020. Vol.98. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE EFFECT OF IMAGE PREPROCESSING TECHNIQUES ON CONVOLUTIONAL NEURAL NETWORK-BASED HUMAN ACTION RECOGNITION

NADYA A SIMANJUNTAK[1], JANOE HENDARTO[2], WAHYONO [3*]

[1] Computer Science Undergraduate Program, Universitas Gadjah Mada, Indonesia

[2,3] Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia

E-mail: [1]nnadya.avirianta.simanjuntak@mail.ugm.ac.id, [2]jhendarto@ugm.ac.id, [3]wahyo@ugm.ac.id
(*corresponding author: Wahyono)

ABSTRACT

The number of the world's population aged 65 or over (elderly) is projected to increase to almost 1.5 billion by 2050. Elderly is vulnerable to various risks on their daily activities, so it is necessary to recognize their actions with machine vision technology automatically. One of the methods to do action recognition is using Convolutional Neural Network (CNN). However, using CNN without preprocessing will result in poor classification accuracy. The preprocessing methods affect the performance of the resulting model. Therefore, it is necessary to research various image preprocessing methods on CNN input to get the optimal model. In this study, various preprocessing methods, namely resizing, enhancement, creation of binary and gradient images, and data augmentation, are compared. After that, the obtained models are evaluated using action recognition dataset. In the validation results, it is found that the best preprocessing method is 64×64 grayscale image preprocessing with sharpening and augmentation in the form of the horizontal flip, which achieves an accuracy of 0.852. Meanwhile, in the testing results, the preprocessing method that produces the best accuracy is the 64×64 grayscale image preprocessing with sharpening, with an accuracy of 0.660.

Keywords: *CNN, Image Preprocessing, Human Action Recognition, Machine Vision.*

1. INTRODUCTION

The number of the world's population aged 65 or over (elderly) is projected to increase from around 524 million in 2010 to nearly 1.5 billion in 2050 [1]. This is caused by the decline in growth rates and the increase of life expectancy of the population. Many elderly people have lost their ability to live independently due to limited mobility and decreased physical or cognitive function. Aging causes a decrease in some physiological functions of the human body, which makes the elderly vulnerable to various risks when engaged in daily life [2]. It causes their activities to need to be monitored. If all family members are busy working, an elderly caregiver is necessary, but caregivers cannot always be there at all time to supervise, and caregiver fees are quite expensive [3].

To deal with these social changes, solutions like Ambient Assisted Living (AAL) is needed. AAL can be defined as the use of information and communication technology (ICT) in a person's daily life and work environment to enable them to stay active longer, stay socially connected, and live independently until old age. AAL is expected to extend the time someone can live independently in their chosen environment [4], and also monitor the behavior and health status of the inhabitants [5].

Technological developments in the field of digital image processing and machine vision provide various benefits in various fields, including monitoring activities in AAL. Vision-based AAL system can help the elderly and people with disabilities. The vision-based action recognition can monitor the habits and routines of the elderly. Action recognition can also detect abnormal behavior such as falls. Although accidents such as falls cannot be completely avoided, a system that can monitor actions can save lives if it can accurately recognize incidents and produce an immediate alert [6]. The automatic action recognition is expected to minimize the risks faced by the elderly.

Journal of Theoretical and Applied Information Technology
31st August 2020. Vol.98. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

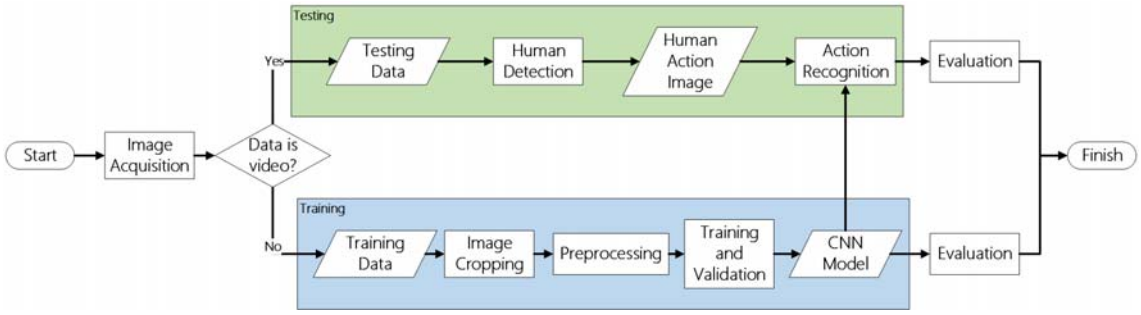

Figure 1: Action recognition flowchart

Several studies have been carried out for action recognition to be utilized in the AAL environment. Study [7] detected abnormal behavior by using Binary Silhouette Average Energy Image (AEI), Histogram of Oriented Gradients (HOG), Principle Component Analysis (PCA), and classification with Support Vector Machine (SVM). Research [8] conducted action recognition to help people with dementia carry out daily activities using the K-Nearest Neighbor based method. [9] utilized action recognition to detect falls using Convolutional Neural Network (CNN) with the LeNet architecture.

According to [10], there is no universal best feature and method descriptor for all datasets in action recognition. However, most action recognition using CNN and deep neural network methods have higher accuracy results compared to action recognition using only handcrafted features. For example, in [11] which used the Bag of Words method, the mean average precision (mAP) for the Pascal VOC dataset is 56.3%. In [12], using CNN, mAP 90.6% is obtained for the same dataset.

According to [13], training on CNN with raw images without preprocessing will cause poor classification performance. Some studies preprocess input images before they are used for the CNN input, such as [14] that used Binary Motion Image as input images into CNN. Research [15] used handcrafted features, namely local binary pattern (LBP) on CNN to classify facial expressions. Better accuracy is obtained than using ordinary CNN because the knowledge that has been processed will be easier to learn and understand by CNN so that recognition is better.

Several studies have compared various preprocessing methods on CNN. Research [13] compared the image preprocessing techniques, namely mean normalization, standardization, and Zero Component Analysis for image classification using the CIFAR10 dataset. Research [16] showed images with preprocessing methods produce better accuracy than images without preprocessing for CNN in the case of emotion recognition. [17] compared the preprocessing techniques for the MNIST dataset, and the results of elastic preprocessing and rotation increased the accuracy by 0.71%.

Preprocessing methods that are used affect the performance of the model generated for a particular problem. From the references obtained, no research which compares various methods of image preprocessing on CNN for action recognition problem has been found. Therefore, research for the preprocessing method on CNN to get the optimal model for action recognition problem is needed. The preprocessing methods to be compared are image preprocessing in the form of RGB, grayscale, binary, and gradient images.

2. RESEARCH METHODS

2.1 The Proposed Method

In this research, there will be a comparison of several images preprocessing methods on CNN for human action recognition. The human action recognition algorithm using CNN is designed based on the flowchart in Figure 1. First of all, a dataset of five classes of action will be collected, namely standing, falling, bending, sitting, and walking. The dataset is taken from images and video from a static camera and only contains one human per image. The data that has been collected will be separated into data for training and testing.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 2: Example of data training

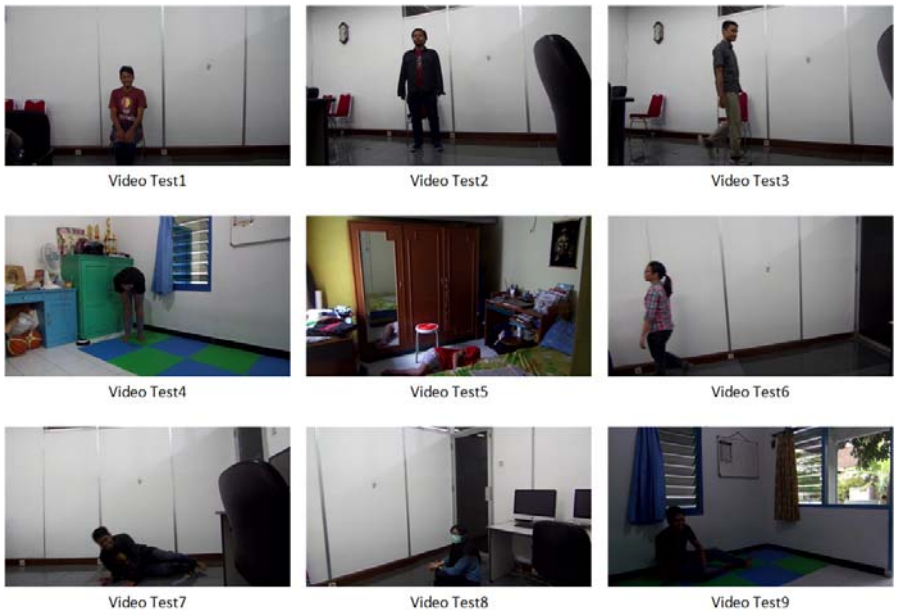

Figure 3: Example of data testing.

At the training stage, human detection is not carried out. The cropping is manually done to get human images that will be used as input on CNN. After cropping, several of image preprocessing scenarios will be carried out. Preprocessing methods that will be compared for CNN input including image enhancement, converting RGB image to grayscale, creating binary image, and calculating gradient image. After that, there will be a process of resizing and normalization. After preprocessing, the image will be used for CNN input. Data augmentation will be carried out before the training process.

In the testing stage, human detection will be performed by doing foreground extraction using the gaussian mixture model method. After the foreground is obtained, a bounding box is created. After that, cropping will be done based on the bounding box. Preprocessing is performed on the cropped image. Then the image will be classified using the CNN model that has been obtained. The testing data is in the form of a video, and the frame is taken every 0.5 seconds for action recognition. For example, if there is a 5 second video, there will be ten frames, each of which will be recognized. Then, the action class that appear most often from the classification results of each frame become the classified action. The results are then compared with ground truths to determine the evaluation metric. The evaluation used will be k-fold cross-validation. Evaluation also determines TP, FP, TN, FN values to create confusion matrix. Then the accuracy, precision, recall, and F-measure are calculated.

2.2 Dataset

The dataset used is grouped into five action classes, namely standing, sitting, walking, fall, and bend. Each class contains 250 images for training and validation and ten action videos for testing. Example of the dataset used for training can be seen in Figure 2. Example of the dataset used for testing is shown in Figure 3. The dataset used for training came from the subset of Weizmann dataset [18], Stanford40 [19], KTH [20], and UR Fall Detection

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Dataset [21]. Data specification for training is as follows:

1) Standing class: 250 images from Google and Stock Image.
2) Walking class: 4 images from the Weizmann dataset, 3 images from the UR Fall Detection Dataset, 67 images from the KTH dataset, and 176 images from Google and Stock Image.
3) Sitting class items: 9 images from UR Fall Detection Dataset, 16 images from Stanford dataset (reading class), 25 images from Stanford40 dataset (watching TV class), and 200 images from Google and Stock Image.
4) Falling class: 53 images from UR Fall Detection Dataset and 197 images from Google and Stock Image.
5) Bending class: 5 images from the Weizmann dataset, 5 images from the UR Fall Detection Dataset, and 240 images from Google and Stock Image.

The dataset for testing is in the form of video taken from a static camera and containing only one human per image. The test data includes ten action videos, each of which is 1-2 minutes long.

2.3 Preprocessing

There are 8 different preprocessing scenarios that are done in this research. The details are as follows:

1) Preprocessing scenario 1: preprocessing is conducted on human images in the form of RGB and grayscale images by doing resizing and normalization. There is no data augmentation nor enhancement.
2) Preprocessing scenario 2: preprocessing is conducted on human images in the form of RGB and grayscale images by doing resizing and normalization. After that, there will be enhancement process.
3) Preprocessing scenario 3: preprocessing is conducted on human images in the form of RGB and grayscale images by doing resizing and normalization. After that there will be data augmentation.
4) Preprocessing scenario 4: preprocessing is conducted on human images in the form of RGB and grayscale images by doing resizing and normalization. After that, there will be enhancement process and data augmentation.
5) Preprocessing scenario 5: preprocessing is conducted on human images in the form of creating binary images from grayscale images with resizing and normalization.
6) Preprocessing scenario 6: preprocessing is conducted on human images in the form of calculating gradient images from grayscale images with resizing and normalization.
7) Preprocessing scenario 7: preprocessing is conducted on human images in the form of creating binary images from grayscale image with resizing and normalization. After that, there will be data augmentation.
8) Preprocessing scenario 8: preprocessing is conducted on human images in the form of calculating gradient image from grayscale image with resizing and normalization. After that there will be data augmentation.

2.4 Normalization

Data normalization is done to ensure that each input parameter (pixel) has a similar data distribution. Data normalization is done by dividing the value of each pixel by 255 (according to intensity range for RGB images and grayscale), so that the range of intensity values becomes [0,1].

$$Z_i = \frac{X_i}{255} \quad (1)$$

where $(X=X_i, \ldots X_n)$ is original data and Zi is ith normalized data.

2.5 Data Augmentation

Data augmentation is done in real-time during training on CNN. Data augmentation can provide many variations to CNN and increase the amount of relevant data. Augmentation is performed randomly with the ImageData- Generator function from the Keras library. At each epoch, each batch of training images will be transformed by the function of the augmentation method, then images that have been randomly transformed will be used for training so that at each epoch, CNN will conduct training from different data. The data augmentation methods are as follows:

1) Horizontal flip: reverses the image horizontally.
2) Zooming: zooms on a scale from 0.9 to 1.5.
3) Vertical and horizontal shift: shift horizontally to 0.2 percent of the image and make vertical shift with the range of [-10,10] pixels.
4) Rotation: rotates at an angle from 0 to 15 degrees.

2.6 Parameters Tuning

Some preprocessing parameters are tested to get the best parameter for each preprocessing method:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

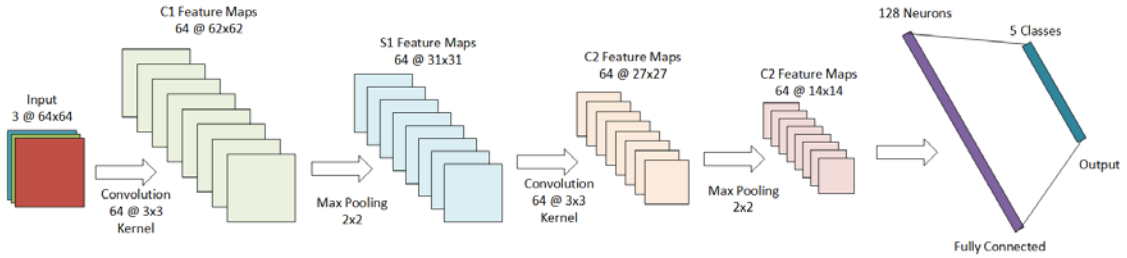

Figure 4: CNN Architecture.

1) The RGB and grayscale images with and without image enhancement are compared. The image enhancement compared are blurring and sharpening.
2) In the gradient image, Sobel and Robert operators to find image gradient are compared.
3) In binary images, the threshold parameters used to create the binary image are 100, 150, and 200.
4) In resizing, three input image sizes will be compared, 32×32, 48×48, and 64×64.
5) Four types of data augmentation will be compared namely zooming, horizontal flip, translation, and rotation.

2.7 CNN Architecture

The CNN architecture used for action recognition consists of the input layer, two convolution layers, two max-pooling layers, fully connected layer, and output layer [22]. Using the training data without preprocessing steps, hyperparameter tuning is done. The CNN architecture for RGB image is shown in Figure 4.

2.7.1. Input Layer

The input size for RGB image is 64×64 and for grayscale image is 32×32. For RGB image, CNN uses three channels. For grayscale, binary, and gradient image, one channel is used.

2.7.2. Convolution Layer 1

The input image is convolved using combinations of 32 and 64 kernels with the size of 3×3 and 5×5 and one stride. The best parameter is 32 kernels with size 3×3 for grayscale image and 64 3×3 kernels for RGB image.

2.7.3. Pooling Layer 1

Pooling Layer The max-pooling kernel size is 2×2 with one stride. The result is to reduce the feature map.

2.7.4. Convolution Layer 2

Convolution uses combinations of 32 and 64 kernels with the size of 3×3 and 5×5 and one stride. The best parameter is 32 kernels with size 3×3 for grayscale image and 64 3×3 kernels for RGB image.

2.7.5. Pooling Layer 2

Max-pooling with 2×2 kernel is performed to reduce the feature map.

2.7.6. Full Connected Layer

Features are flattened and 128, 256, or 512 neurons are produced. The best hyperparameter is 128 neurons. The fully connected layer then produces 5 neurons.

2.7.7. Output Layer

The 5 neurons represent five classes of action. The output of this is 5 probability scores for a particular class to appear. The neuron with the highest probability becomes the classified action for the input image.

To train the CNN, 50 epochs is used. Hyperparameter tuning is done for learning rate, batch size, and dropout rate. The combination of learning rates used are 0.001, 0.002, 0.005, 0.01, 0.05, and 0.1. The size of batch used is 4, 8, 16, 32, and 64. The dropout rate used are 0, 0.1, 0.2 and 0.5. The best parameters for RGB images are the batch size 64, learning rate 0.001, and dropout rate 0.1. The number of kernels in the first and second convolution layer is 64 with size 3×3. The best number of neurons in the dense layer is 128. The best parameters for grayscale image are batch size 64, learning rate 0.002, and dropout rate 0.2. The number of kernels in the first and second convolution layer is 32 with size 3×3. The best number of neurons in the dense layer is also 128.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Resizing and normalization effect results

Image Type-Size	Image Size	Loss	Accuracy
RGB	32×32	0.741	80.90%
RGB	48×48	0.781	80.50%
RGB	64×64	0.667	81.30%
Grayscale	32×32	0.713	79.80%
Grayscale	48×48	0.878	76.60%
Grayscale	64×64	0.715	79.80%

Table 2: Enhancement effect results

Image Type	Image Size	Enhancement	Loss	Accuracy
RGB	32×32	Blurring	0.693	78.50%
RGB	48×48	Blurring	0.761	77.70%
RGB	64×64	Blurring	0.761	77.70%
RGB	32×32	Sharpening	0.674	79.40%
RGB	48×48	Sharpening	0.762	78.20%
RGB	64×64	Sharpening	0.529	81.40%
Grayscale	32×32	Blurring	0.671	78.40%
Grayscale	48×48	Blurring	0.683	79.10%
Grayscale	64×64	Blurring	0.725	79.40%
Grayscale	32×32	Sharpening	0.769	76.60%
Grayscale	48×48	Sharpening	0.747	79.80%
Grayscale	64×64	Sharpening	0.604	83.20%

Table 3: Augmentation effect results

Image Type	Augmentation	Loss	Accuracy
RGB	Zooming	1.182	73.00%
RGB	Horizontal Flip	0.669	78.00%
RGB	Rotation	0.655	80.40%
RGB	Translation	1.734	53.80%
RGB	Flip with Rotation	0.782	75.80%
Grayscale	Zooming	0.744	79.70%
Grayscale	Horizontal Flip	0.625	82.00%
Grayscale	Rotation	0.725	79.00%
Grayscale	Translation	1.873	51.20%
Grayscale	Flip with Rotation	0.824	78.60%

3. EXPERIMENTAL RESULTS

3.1 Preprocessing Effect

In this section, we evaluate the effect of preprocessing on the input image before going through CNN network. The evaluations include resizing, normalization, enhancement, and data augmentation. All the processing strategies will be evaluated regarding to the loss score and accuracy of the proposed system.

3.1.1. Resizing and Normalization

This preprocessing method is done for RGB and grayscale images as can be seen in Table 1. For RGB image, the larger the image size, the accuracy is higher. For grayscale image, the larger the image size, the accuracy is getting lower. The best parameter for this method is RGB image with size 64×64 and grayscale image with size 32×32.

3.1.2. The Effect of Image Enhancement

For both RGB and grayscale images, we evaluate the effect of blurring and sharpening. As shown in Table 2, it is found that in both image type, the sharpening enhancement gains better accuracy comparing to blurring enhancement approach. The best parameter for both RGB and grayscale image is image with size 64×64 and sharpening as the enhancement method.

3.1.2. The Effect of Data Augmentation

For this preprocessing scenario, data augmentation is done after doing resizing and normalization for both RGB and grayscale images. This data augmentation includes zooming, horizontal flip, rotation, translation, and flip with rotation. Table 3 shows the accuracy obtained from

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Binary Image effect results

Threshold	Image Size	Loss	Accuracy
100	32×32	0.787	75.80%
100	48×48	0.774	78.20%
100	64×64	0.767	75.80%
150	32×32	0.802	76.40%
150	48×48	0.788	76.10%
150	64×64	0.794	78.00%
200	32×32	0.842	72.90%
200	48×48	0.882	74.60%
200	64×64	0.853	76.20%

Table 5: Binary Image with augmentation effect results

Augmentation	Loss	Accuracy
Zooming	0.924	75.60%
Horizontal Flip	0.793	77.10%
Rotation	0.841	78.40%
Translation	2.186	37.10%
Flip with Rotation	0.891	74.70%

Table 6: Gradient image effect results

Operator	Image Size	Loss	Accuracy
Robert	32×32	0.739	79.00%
Robert	48×48	0.810	79.30%
Robert	64×64	0.703	80.20%
Sobel	32×32	0.945	73.10%
Sobel	48×48	0.983	72.60%
Sobel	64×64	1.477	69.40%

Table 7: Gradient Image with augmentation effect results

Augmentation	Loss	Accuracy
Zooming	0.747	78.90%
Horizontal Flip	0.943	78.30%
Rotation	0.663	81.20%
Translation	2.181	45.40%
Flip with Rotation	0.983	72.40%

these data augmentation strategies. For RGB image, the augmentation method that produces the highest validation accuracy is rotation. For grayscale, the highest accuracy is obtained with horizontal flip as the augmentation method.

3.2 Preprocessing on Input Image

Beside RGB and grayscale images, we also conducted experiment on binary and gradient images as input in the convolutional network.

3.2.1. Binary Image as Input

In this scenario, creation of binary image from the grayscale image is done. Several thresholds to make the binary image are compared. The best threshold is 150 and the image size is 48×48. For this image size, if the threshold is getting larger, the accuracy is getting lower. The result of this preprocessing scenario is shown in Table 4.

The next scenario is doing data augmentation for the binary image. The best augmentation method that produces the highest accuracy for the binary image is rotation. The result of this preprocessing scenario is shown in Table 5.

3.2.2. Gradient Image as Input

In this scenario, creation of gradient image from the grayscale image is done. Two operators, Robert, and Sobel, to make the gradient images are compared. The best operator is Robert and the image size are 64×64. The result of this preprocessing scenario is shown in Table 6.

The next scenario is doing data augmentation for the gradient image. The best augmentation method that produces the highest accuracy for the gradient image is rotation. The result of this preprocessing scenario is shown in Table 7.

3.3 Evaluation on Data Validation

This part compares the accuracy of the best model obtained in subsection 3.1 dan 3.2. For this purpose, we conducted the k-fold cross validation using validation data. The results of k-fold cross validation from the best model of each preprocessing scenario are shown in Table 8. From Figure 8, it can be seen that all models with augmentation produce better accuracy than those without augmentation. The highest accuracy, precision, recall, and f-measure scores are obtained from the grayscale image model with sharpening as enhancement and horizontal flip as augmentation. The accuracy obtained is 0.853. This model increases the grayscale model with enhancement but without augmentation by 0.023. The accuracy is 0.045 higher than grayscale model without enhancement or augmentation. The accuracy and loss of the best model can be seen in Figure 5.

3.4 Evaluation on Data Testing

During the testing stage, the test data used are 10 action videos, each of which had a duration of 1-2 minutes. After human detection is carried out, action recognition is done using CNN models that have been obtained from the training process. Figure 6 shows an example of the action recognition results from each test video.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Evaluation result on validation data

Model	Accuracy
RGB	80.10%
RGB with Augmentation	80.40%
RGB with Enhancement	81.40%
RGB with Augmentation and Enhancement	82.10%
Grayscale	79.80%
Grayscale with Augmentation	82.00%
Grayscale with Enhancement	83.20%
Grayscale with Augmentation and Enhancement	85.30%
Binary	78.20%
Binary with Augmentation	78.40%
Gradient	80.20%
Gradient with Augmentation	81.20%

Table 9. Evaluation result on testing data

Model	Accuracy
RGB	30.20%
RGB with Augmentation	25.50%
RGB with Enhancement	34.00%
RGB with Augmentation and Enhancement	28.30%
Grayscale	34.00%
Grayscale with Augmentation	34.90%
Grayscale with Enhancement	66.00%
Grayscale with Augmentation and Enhancement	28.30%
Binary	38.70%
Binary with Augmentation	18.90%
Gradient	41.50%
Gradient with Augmentation	28.30%

A comparison of the results of the action recognition evaluation is shown in Table 9. The grayscale model with enhancement produces the highest accuracy, precision, recall, and f-measure compared to other models. The resulting accuracy is 0.660. Although the grayscale model with enhancement and augmentation produces the highest accuracy in k-fold cross-validation, with test data, the accuracy of only 0.283 is obtained. In the test data, for all models, except the grayscale model, data augmentation makes the accuracy decreases.

The highest average accuracy of all 10 videos is obtained from video 10. The accuracy score is 0.508. The placement of the camera makes video 10 has no occlusion at all, the lighting is good enough, and every action can be recognized properly. For the grayscale model with enhancement, video 10 can produce 1.00 accuracy, which means every action is correctly recognized.

The lowest average accuracy of all 10 videos is obtained by video 8, with an accuracy of only 0.220. This is because video 8 has a lot of occlusions, so the actions performed cannot be adequately recognized. The grayscale model with enhancement produces a pretty high accuracy of 0.818 for this video. However, other models cannot classify actions properly. All other models' accuracy scores are below 0.3.

Figure 7a shows an error in the action recognition due to the occlusion and similarity of the clothing's color to the subject. The action class should be standing class, but the model recognizes it as sitting class. Occlusion occurs at the bottom of the body, so the subject's legs are not visible. The skirt that the subject wears is black, and the color is similar to the background so that it is not detected as foreground. Human detection only partially detects the body of the subject.

Figure 7b points out the error in action recognition from standing class to walking class. This error occurs because of the similarity between the action walking and standing. The resulting model sometimes classifies walking classes as standing and vice versa. Figure 7c indicates an error in action recognition due to poor lighting. Human detection methods cannot detect the whole body of the subject. An error occurred from a sitting class that is classified as a walking class.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

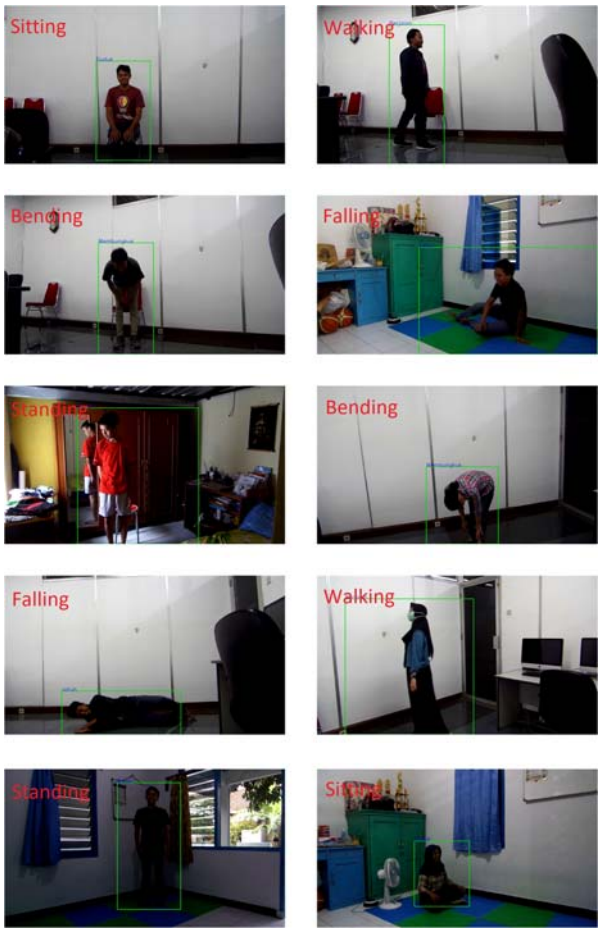

Figure 6: The sample results of action recognition from testing videos with detected area is indicated by green bounding boxes.

(a) Due to occlusion (b) Due to action similarity (c) Due to poor lighting

Figure 7: Sample of Recognition Errors.

The confusion matrix for the k-fold cross validation result for a grayscale model with sharpening and augmentation can be seen in Figure 8. The highest classification accuracy is on the standing and fall class. Confusion matrix of testing video for grayscale with sharpening model is shown in Figure 9. The highest classification accuracy is on the walking class. A lot of misclassifications happen in the sitting class.

4. CONCLUSION

In this paper, we have implemented the proposed method to make a comparison between preprocessing methods on CNN for human action recognition for five action classes, namely sitting, standing, walking, fall, and bend. Based on the k-fold cross validation result, the best preprocessing method is 64×64 grayscale image preprocessing with sharpening and augmentation in the form of the horizontal flip, with an accuracy of 0.852. In the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

testing results, it is found that best accuracy of 0.660 was obtained when utilizing the 64×64 grayscale image preprocessing with sharpening. Overall, it can be concluded that the preprocessing stage give effect to the accuracy of human action recognition system. In most cases, preprocessing results improve the accuracy.

Some of the misclassifications of the test dataset occur because of occlusion, lighting changes, clothing variations, and human detection errors. These problems will be open issues to be considered in the future works. To further improve the resulting model, research needs to be done using temporal information for action recognition, not just spatial information. The model that produces the best accuracy for testing is grayscale image processing with sharpening. However, this model still cannot recognize all actions correctly, so there is a need for further research on models and other preprocessing methods that can improve the accuracy of action recognition. It is also necessary to use more and diverse training data with more variations in age, gender, lighting, camera position, and clothing used. The use of Gaussian Mixture Model is a human detection method is still unable to overcome the problem of occlusion and lighting changes, so further research is needed using other human detection methods that are more resistant to occlusion and lighting.

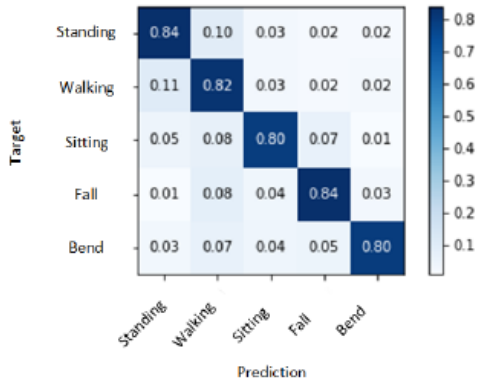

Figure 8: Confusion matrix for k-fold cross validation

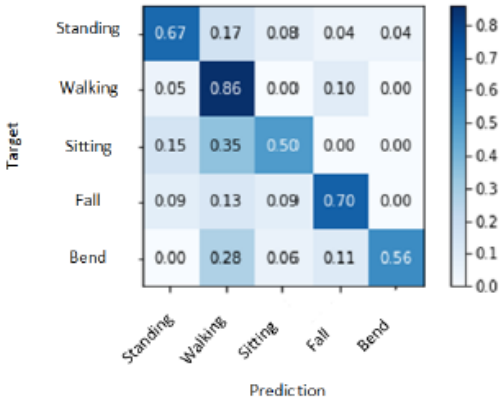

Figure 9: Confusion matrix for testing video

REFRENCES:

[1] World Health Organization, "Global Health and Aging," *World Health Organization*, Tech. Rep., 2011.

[2] R. Yared and B. Abdulrazak, "Ambient Technology to Assist Elderly People in Indoor Risks," 2016.

[3] C.-d. Huang, C.-y. Wang, and J.-c. Wang, "Human Action Recognition System for Elderly and Children Care Using Three Stream ConvNet," no. 4, pp. 5–9, 2015.

[4] D. Calvaresi, D. Cesarini, P. Sernani, A. Franco, and D. Arnon, "Exploring the ambient assisted living domain: a systematic review," *Journal of Ambient Intelligence and Humanized Computing*, 2016.

[5] S. Colantonio, G. Coppini, D. Giorgi, M.-a. Morales, and M. A. Pascali, Computer Vision for Ambient Assisted Living Monitoring Systems for Personalized Healthcare and Wellness That Are Robust in the Real World and Accepted by Users , Carers , and Society. Elsevier Ltd, 2018.

[6] K. Adhikari, H. Bouchachia, and H. Nait-Charif, "Activity Recognition for Indoor Fall Detection Using Convolutional Neural Network," *in 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, Nagoya, 2017.

[7] D. Lahiri, C. Dhiman, and D. K. Vishwakarma, "Abnormal human action recognition using average energy images," *in 2017 Conference on Information and Communication Technology*, CICT 2017, vol. 2018-April, pp. 1–5, 2018.

[8] E. M. Jean-Baptiste and A. Mihailidis, "Benefits of automatic human action recognition in an assistive system for people with dementia," *in IHTC 2017 - IEEE Canada International Humanitarian Technology Conference 2017*, pp. 61–65, 2017.

[9] A. Akula, A. K. Shah, and R. Ghosh, "Deep learning approach for human action recognition in infrared images," *Cognitive Systems Research*, vol. 50, pp. 146–154, 2018.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] A. Sargano, P. Angelov, and Z. Habib, "A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition," *Applied Sciences*, vol. 7, no. 1, p. 110, 2017.

[11] V. Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," *Advances in Neural Information Processing Systems*, pp. 1–9, 2011.

[12] X. Wan, K. Li, and Y. Li, "A Deep Model Combining Structural Features and Context Cues for Action Recognition in Static Images," *Journal of the Society of Mechanical Engineers*, vol. 90, no. 823, pp. 758–759, 2017.

[13] K. K. Pal and K. S. Sudeep, "Preprocessing for Image Classification by Convolutional Neural Networks," *in IEEE International Conference on Recent Trends in Electronics Information Communication Technology*, 2016, pp. 1778–1781.

[14] T. Dobhal, V. Shitole, G. Thomas, and G. Navada, "Human Activity Recognition using Binary Motion Image and Deep Learning," *Procedia Computer Science*, vol. 58, pp. 178–185, 2015.

[15] S. Sawardekar, P. Sowmiya, and R. Naik, "Facial Expression Recognition using Efficient LBP and CNN," no. June, pp. 2273–2277, 2018.

[16] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, "Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition," *Procedia Computer Science*, vol. 116, pp. 523–529, 2017.

[17] S. Tabik, D. Peralta, A. Herrera-Poyatos, and F. Herrera, "A snapsot of image pre-processing for convolutional neural networks: case study of MNIST," International J*ournal of Computational Intelligence Systems*, vol. 10, no. 1, p. 555, 2016.

[18] L. Gorelick, M. Blank, E. Shechtman, S. Member, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[19] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," *in Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1331–1338.

[20] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," *in Proceedings 17th International Conference on of the Pattern Recognition* Volume 3 - Volume 03. Washington, DC, USA: IEEE Computer Society, 2004, pp. 32–36.

[21] M. Kepski and B. Kwolek, "Fall Detection on Embedded Platform Using Kinect and Wireless Accelerometer," *Computer Methods and Programs in Biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.

[22] N. A. Simanjuntak, "Comparison of Preprocessing Technique on CNN for Human Action Recognition," B.Comp. thesis, Dept. Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia, 2019.