INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 9, Issue 2 - V9I2-1142)
Available online at: https://www.ijariit.com

Image classification of human action recognition using transfer learning in PyTorch

Amos Oyetoro
oyetoro.amos@gmail.com
Austin Peay State University, Clarksville, United States

ABSTRACT

Over the years, deep learning models have been applied to human action recognition (HAR). due to the enormous amount of labeled data needed to train deep learning models, there has been a significant delay in the absolute development of these models. Data collection in sectors like HAR is challenging, and human labeling is expensive and time-consuming. The current approaches mainly rely on manual data gathering and accurate data labeling, which is carried out by human administration. This frequently leads to a slow and prone to human bias labeling data collection method. To solve these issues, we offered a novel approach to the current data collection techniques [1]. It is generally used that (CNN) is among machine learning models. Since Yann Lecun created this context in 1988, image identification has greatly improved. Transfer learning in image classification has simplified the process of training new models from the beginning and has reduced the number of data points that need to be processed, it was used in this project to classify human actions.

Keywords: *Transfer learning, CNN, Model Pre-trained, Precision, Recall, PyTorch, Pandas, NumPy, Seaborn, Matplotlib, Scipy, ResNet18, Confusion Matrix*

I. INTRODUCTION

Vision-based and sensor-based methods make up Human Recognition System, and these methods are further divided into wearables, object-tagged, and dense sensing methods. In addition to these design issues, there are some issues related to Human action recognition systems, such as the type of sensors they use, data collection rules, how quickly they recognize changes, the amount of energy they consume, the processing speed, and the flexibility they provide. To design a human activity recognition model that is effective and lightweight, all parameters must be considered [2]. For human action recognition, a long-short memory approach has been proposed using datasets from kaggle.com.

Analyzing this data can be a big task due to the intricacy of human activities and the existing disparities between two individuals As a special framework suggested by Yann Lecun, CNN is part of numerous algorithms introduced. This architecture improved significantly in image identification although among some little concerns one of which is the time to train a model especially if you have very large datasets. Using a pre-trained convolutional neural network helped us identify the type of human action using a large data set. It is essential to utilize a model which reduces the time required to train the model. [23].
Among the benefits of the prototype are:

Prepare a network based on large datasets and load it with pre-trained weights.
A convolutional layer can be frozen by freezing all the weights: the weights should be adjusted to fit the similarity of the new task to the original dataset.
Adding a classifier at the top of the network: The number of outputs should equal the number of classes.
Optimize the model for a smaller dataset by training only the custom classifier layers.

II. THEORETICAL FRAMEWORK

The style that would be used in this project would consist of the use of PyTorch to design the algorithm. Among its advantages are its ability to control all aspects of model development, its ability to implement back transmission with tensor auto differentiation, its speed, and its ease of debugging due to the dynamic nature of PyTorch graphs. Object recognition transfer learning outline. A large data set can be used to train a pre-trained model.

To store relevant variables, a configuration script is needed.
Providing helper functions for dataset loaders
Our dataset on disk needs to be built and organized so that PyTorch's Image Folder and Data Loader classes are easily accessible.
A CNN model is trained on a huge dataset using a pretrained algorithm.
The script extracts feature from a driver to perform basic transfer learning.
The analysis of features and the fine-tuning of hyperparameters.
An additional driver script that fine-tunes the fully connected (FC) layer of a pre-trained network by replacing it with a fresh, newly generated FC header.

Deep learning is an approach to machine learning that involves learning from a large amount of data by using multilayered neural networks which were used in this project. As shown in figure 1.
A deep learning algorithm mimics the way humans gain knowledge by using AI (artificial intelligence). It is imperative that deep learning be incorporated into data science, along with statistics and predictive modeling. It helps in data collection, analysis, and interpret large amounts of data, which was implemented in this project, deep learning is extremely helpful; it accelerates and simplifies the process.

Convolutional neural networks are important parts of neural networks, computer vision works with Convolutional neural networks and dealing learning.

In CNN, an image is received and weighted based on the different objects in it, and then the different objects are distinguished. Compared to other deep learning algorithms, CNN requires a very small amount of data preprocessing. CNN can learn characteristics about a target object thanks to its fundamental methods for training its classifiers. CNN shares a similar architecture with neurons in the brain. A particular example is a Visual Cortex. There are a variety of modules that follow a set workflow in a CNN algorithm. Here are a few examples.

Image to be input.
The convolution layer (Kernel)
The pooling layers
An overview of the fully connected layer of classification
The architecture

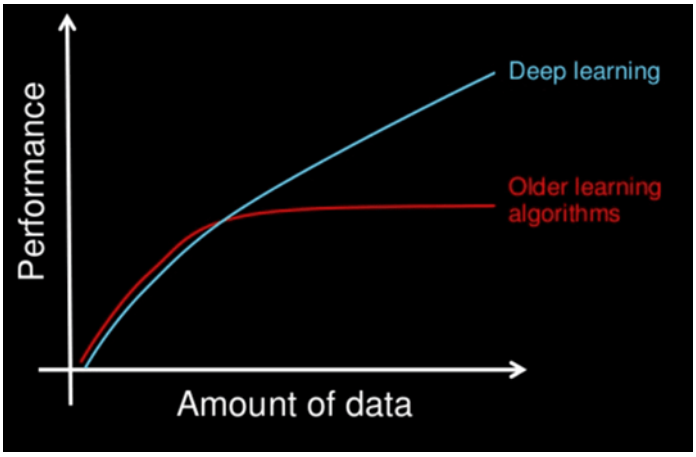


Figure 1: This show how data is been scaled using deep learning

III. BACKGROUND OF THE PROJECT

An extensive study has been conducted on human action recognition for many years. An important portion of the action of concern in an image is manually explained by most action recognition methods. Researchers have observed that the action of interest can be identified spontaneously by identifying the relevant portion [4].
One of the highly functional fields of computer vision research is human behavior analysis, which uses information from images or

series of images to determine human behavior [5]. Images captured by conventional cameras cannot be analyzed comprehensively. Existing research primarily focuses on detecting body parts and estimating poses [7]. Intensity images are used in research to recognize human actions [6]. Besides the complexity of human actions, the diversity of the human body, appearance, posture, motion, clothing, illumination, and view angle makes it very difficult to automatically recognize human actions [8].

Techniques based on feature extraction and trajectory tracking have less discriminative power than handcrafted methods. Deep systems are, on the other hand, inefficient for capturing salient motions [9]-[10]. Combining a trajectory for action recognition with deep convolutional networks helps address this issue [1]. However, Additionally, there are some limitations inherent to deep learning-based approaches. These models require huge datasets for training, and many domain-specific data sets are expensive and time-consuming to collect. In a domain-specific problem, it is therefore not feasible to train a deep-learning algorithm from scratch. With transfer learning, it is possible to use an existing small dataset can be used to train the objective algorithm using a network as a resource architecture [11]

Researchers are devoted to identifying and segmenting HAR, although very few have reported it. There have been a variety of methods used for classification in recent years, including K-Nearest Neighbor (KNN) [12], and SVM classifiers (SVMs), The most accurate classification technique has been deep learning around 100% correctness for cycling from datasets compared to decision trees and random forests [13-16].

Additionally, some pre-trained models exist, including DenseNet [19] and [20], Google Net, and Resnet [21] and [23].

IV. DATA ANALYSIS

Looking at the data analysis, this data was gotten from the Kaggle website which has about (12,600) images that show the activities performed by humans which are categorized into:

That comprises 840 callings, 840 clappings, 840 cyclings, 840 dancing, 840 drinkings, 840 eating, 840 fighting, 840 huggings, 840 laughings, 840 listening_to_music, 840 running, 840 sitting, 840 sleeping, 840 texting, 840 using_laptop.

This dataset has two stages:

i. Training
ii. Validation

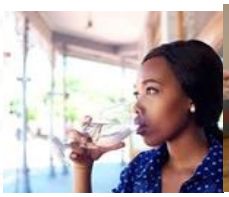
The datasets include 10,710 images divided into 714 images of calling, 714 clapping, 714 cyclings, 714 dancing, 714 drinking, 714 eating, 714 fighting, 714 huggings, 714 laughings, 714 listening_to_music, 714 running, 714 sitting, 714 sleeping, 714 texting, 714 using_laptop

Validation: It entails the following datasets: 1,890 images separated into 126 calls, 126 clappings, 126 cyclings, 126 dancing, 126 drinkings, 126 eating, 126 fighting, 126 huggings, 126 laughings, 126 listening_to_music, 126 running, 126 sitting, 126 sleeping, 126 texting, 126 using_laptop. The sample images from the datasets are shown below:


i. Calling


ii. Clapping


iii. Cycling



iv. Dancing v. Drinking vi. Eating

vii. Fighting viii. Hugging ix. Laughing

x. Listening to music. xi. Running xii. Sitting

xiii. Sleeping xiv. Testing xv. Using laptop

Figure 2: Human action dataset

This dataset has steps to pre-process to make it more useful for the model, there are view steps to apply, which are:
Transform: The data transformations that must be performed, including pre-processing steps and augmentations to the data for the model
Batch size: The number of batches that should be generated from the Data Loader
Shuffle The shuffle of data - during training but not during validation, we will shuffle data.
Resize: The images were built on 240 which has been mentioned in the PyTorch module

V. DESIGN AND METHOD

This study suggests a verifier-designated identity-based remote data integrity checking scheme. This plan can also achieve data privacy protection and resolve the semi-trusted verifier problem. Data owner is unable to access data that has already been shared or data that is shared in the future. Additionally, a concrete IBE construction is shown. The proposed IBE system is demonstrated to be adaptive-secure in the decisional standard model.

Transfer learning has six steps which are:
First, obtain the pretrained model.
Creating a base model
Freezing layers so that it does not change during training the model.
Adding a new trainable layer
Improving the mode via fine-tuning
Trained the new layers on the dataset.
In this model the listed below are the various libraries that are utilized.

Pandas: for loading datasets, this helps in manipulation and analysis, it is an open source.
NumPy: An array object consists of multiple dimensions, and several routines are provided for processing them. Arrays can be mathematically and logically manipulated using NumPy. It also provides an interesting collection of data structures that makes it effective to work easily with various datasets.
Matplotlib: This will be used in creating statics, and animated and graphical plotting for interactive virtualization.
Seaborn: This will be used for beautifying graphs for high-level interfaces.
PyTorch: It is an open-source machine library that is Natural language processing and computer vision applications. The application reduces boilerplate code and follows an object-oriented method when developing an algorithm. Transfer learning is done with a pre-trained algorithm called ResNet18. ResNet18 entails 256 and three-three dimensional kernels converging to provide one layer of

convolution.

SciPy: This will be used for different scientific computations, statistical functions, and image processing.

Sklearn: This is an efficient tool for prediction which will be used in classification, regression, clustering, and dimensionality

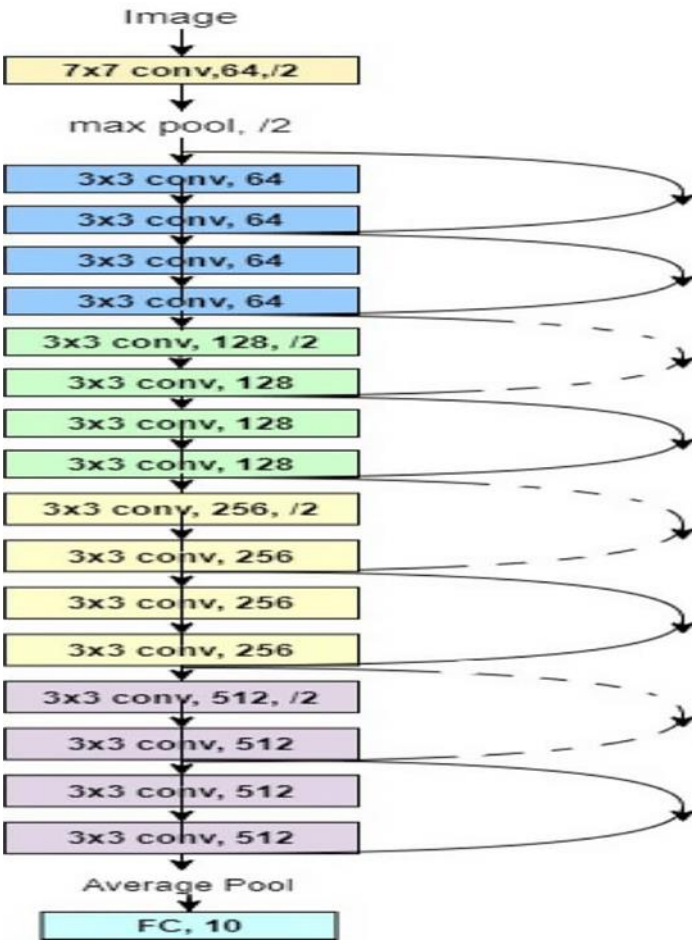


Figure 3: ResNet18 architecture

VI. IMPLEMENTATION

The transform sub-module from torchvison was used to initially downsize the images from 240 X 240 to 224 X 224. Then it was rotated by a range of [-90, 90] and horizontally rotated before being converted into a tensor image. After that, the photos were standardized by applying a mean and standard deviation figure [0.4914, 0.4822, 0.4465]. These appeared in the train and validation data obtained from Kaggle.

Our resnet18 model's components are looped over, and the batch normalization modules' settings are set to just not trainable. The pretrained algorithm is given a new system head and linked to it. This is how our test's fine-tuning procedure worked. The cross-entropy loss can be used to train this new layer in place of a gentle maximum classifier whenever the cross-entropy loss is applied. Using Adam's optimizer, a 0.001 learning value was achieved.

The training is performed across 120 epochs, and after each batch of training data is imported, the model values are modified, the slopes are adjusted to zero, and the loss and gradients are reported. All epoch's training loss is computed as well.

We analyze our model using the validation dataset, and while we're doing so, after that we switch off autograd and leave the algorithm in preview mode. Both the overall loss and the number of reliable verification forecasts are calculated.

VII. DATA ANALYSIS RESULTS

Several parameters, including precision, accuracy, recall, and F1 score, were determined from data analysis results.

As a result of training the model for 120 epochs, 69.95% of the validation accuracy was obtained, and 77.27% of the training accuracy was achieved. We can denote that training with more epochs there is certain that the percentage of the result will keep increasing.

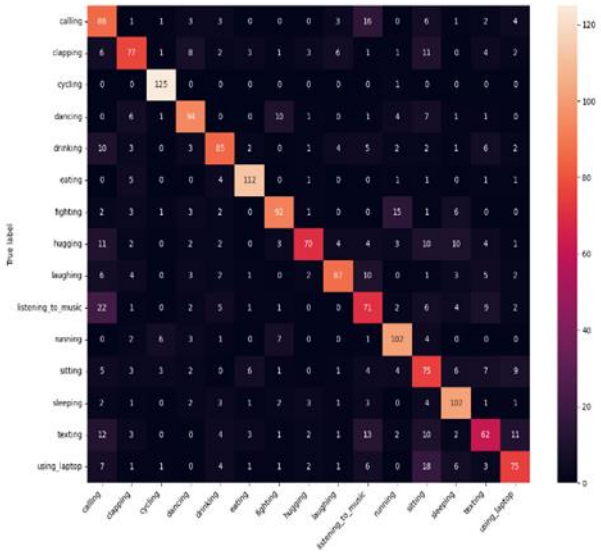


Figure 4: An overview image of the confusion matrix's

Fig 4 shows a confusion matrix from which the following analysis of classification can be extracted.
Chart 1: An analysis of classification Table showed:

ACTIONS	PRECISION	RECALL	F1-SCORE	ACCURACY
calling,	**0.500**	**0.722**	**0.591**	**0.910**
clapping,	**0.686**	**0.643**	**0.664**	**0.942**
cycling,	**0.856**	**0.992**	**0.919**	**1.000**
dancing	**0.764**	**0.770**	**0.767**	**0.923**
drinking	**0.882**	**0.476**	**0.618**	**0.804**
eating	**0.868**	**0.889**	**0.878**	**0.935**
fighting	**0.762**	**0.738**	**0.750**	**0.940**
hugging	**0.886**	**0.556**	**0.683**	**0.754**
laughing	**0.820**	**0.722**	**0.768**	**0.880**
listening_to _music	**0.556**	**0.714**	**0.625**	**0.880**
running	**0.779**	**0.754**	**0.766**	**0.950**
sitting	**0.458**	**0.611**	**0.524**	**0.930**
sleeping	**0.871**	**0.698**	**0.775**	**0.874**
texting	**0.579**	**0.524**	**0.550**	**0.810**
using_laptop	**0.657**	**0.683**	**0.670**	**0.912**

VII. DISCUSSIONS AND CONCLUSIONS

The algorithm is trained over 120 iterations by applying An Adam optimizer and a learning rate of 0.001. The suggested framework had a 77.27% accuracy rate. According to the results, sleeping demonstrated greater precision, but sitting was associated with lower precision., our methodology creates it incredibly simple for non-professional individuals viewing a deep learning algorithm working.

This Recognition of human activity continues to be a significant challenge for computer vision. With the HAR platform, you can build a variety of applications, such as video surveillance, healthcare, and human-computer interface. Technologies and methodologies have advanced significantly over the previous few decades and have continued to do so. However, there are also difficulties when dealing with practical settings in addition to the problem of interclass similarity and innate intraclass diversity.

For both the benchmark datasets and the recognition algorithms, many of the recognition tasks are solved on a case-by-case basis. Combining several datasets into a single, sizable, complicated, and comprehensive dataset is the route research should go in the future. Even if each dataset may serve as a benchmark in its niche, combining them all results in more efficient, universal algorithms that are closer to real-world occurrences. For instance, modern deep learning is reportedly more effective when four larger datasets are pooled.

the project uses ResNet18 as the training model to construct a deep learning model that leverages transfer learning to categorize various human action recognitions. The preprocessing methods of resizing, horizontal flipping, and normalizing are used to improve the model.

It has been shown that pattern recognition systems could be used to automatically classify human physical activity based on on-body accelerometer data. The paper's pursuit of a Markov modeling strategy for the construction of one such machine is a significant impact.

VIII REFERENCES

[1] H.Wang and C Schmid, “Action recognition with improved trajectories,” in Proc.IEEE Int. Conf.Comput. Vis., Dec 2013, pp 3551- 3558.
[2] H. Wang, A .Klaser, C.Schmid and C-L.Liu, “Action recognition by dense trajectories,” in Proc. IEEE Conf. Comput. Vis.Pattern Recog., Jun. 2011, pp 3169-3176.
[3] Y. Jiang, Q.Dai, X.Xue, W.Liu and C-W Ngo. “Trajectory-based modeling of human actions with motion reference points,” inProc. Eur.Conf.Comput.Vis., Oct 2012, Vol 7576, pp.425-438
[4] M. Atiqur A. Rahman, J. Tan, H. Kim, and S. Ishikawa, “Human Activity Recognition: Various Paradigms”, International Conference on Control, Automation and Systems, pp-1896 to 1901, Oct. 14-17, 2008, in COEX, Seoul, Korea.
[5] R. NazliIkizler, C. Gokberk, P. Selen and D. Pinar, “Recognizing Actions from Still Images”, Indian Council of Philosophical Research, ICPR, New Delhi,2007.
[6] T. Lan, Y. Wang, and G. Mori, “Discriminative figure-centric models for joint action localization and recognition,” in Proc. IEEE Int. Conf. Comput. Vis., Nov. 2011, pp. 2003–210.
[7] M. Vrigkas, C. Nikou and A. Kakadiaris (2015) “A Review of Human Activity Recognition Methods”. Front. Robot. AI 2:28. doi: 10.3389/frobt.2015.00028
[8] W. Brendel and S. Todorovic, “Learning spatiotemporal graphs of human activities,” in Proc. IEEE Int. Conf. Comput. Vis., Nov. 2011,
[9] J. Morris, J.R.W. “Accelerometry—a technique for the measurement of human body movements”. J. Biomech 1973, 6, pp729–736.
[10] M. Raptis, I.Kokkinos and S.Soatto,” Discovering discriminative action parts from mid-level video representations”, in Proc, IEEE Conf.Compute.Vis, Pattern Recog., Jun 2012, pp.1242-1249
[11] D. Weinland,” A survey of vision-based methods for action representation, segmentation, and recognition Computer Vision and Image Understanding, 115 (2011), pp. 224-241
[12] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, “Action detection in complex scenes with spatial and temporal ambiguities,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., Sep.–Oct. 2009, pp.128–135.
[13] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in NIPS, 2014.
[14] S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe, and S. Mougiakakou, ‘‘Multisource transfer learning with convolutional neural networks for lung pattern analysis,’’ IEEE J. Biomed. Health Inform., vol. 21, no. 1, pp. 76–84, Jan. 2017.
[15] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, ‘‘Densely connected convolutional networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CPR), Jul. 2017, pp. 2261–2269.
[16] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell and G. Celler” Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring” IEEE Trans. Inf. Technol. Biomed 10 2006 156-167
[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ‘‘Going deeper with convolutions,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CPR), Jun. 2015, pp. 1–9.
[18] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE
[19] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. VenuGopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in CVPR, 2015
[20] G. Huang, Z. Liu, L. Maaten, and K. Weinberger, ‘‘Densely connected convolutional networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CPR), Jul. 2017, pp. 2261–2269.
[21] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE
[22] L.Al, F.Alnajjar, H.Jassmi, M.Gocho, W.Khan, M.Serhani. “Performance Evaluation of Deep CNN-Based Crack Detection

and Localization Techniques for Concrete Structures". Sensors 2021, 21, 1688. https://doi.org/10.3390/s21051688
[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CPR), Jun. 2015, pp. 1–9.

Amos Oyetoro: Obtained his bachelor's degree in Computer Engineering from Nigeria in 2012. And his M.s in Computer Science from Austin Peay State University, Clarksville, Tennessee, United States. He had over eight years working experience, Amos has held various positions in the Information Technology industry, from System development to system design and implementation. He has a strong background in Cloud computing and cyber security, Application and system architecture, database design, web application development, and system analysis. He possesses an active member of multiple scientific organizations such as IEEE, the National Society of Professional Engineers, The Nonprofit Technology Enterprise Network, and the National Society of Black Engineers. His primary interest includes Cloud computing, Network and security, Data Analysis and performance, machine learning, and Information Technologies

