Convolutional Neural Network as an Extractor Feature For Image Search

Sandi Fajar Rodiyansyah[1], Ardi Mardiana[2]
{rodiyansyah@unma.ac.id[1] , aim@unma.ac.id[2] }

Universitas Majalengka, Majalengka, West Java, Indonesia

Abstract. The deep learning method that can be used for image search was a convolutional neural network but there were many parameters and design decisions that were difficult to determine. In this study, a convolutional neural network method was used as an extractor feature for image search. Optimal feature extraction was performed at the last second fully connected layer (FC2) and used cosine distance as a distance metric with a threshold of 0.4. Producing the accuracy of model classification on the test data was 87.72%.

Keywords: deep learning; convolutional neural network; extractor feature; image search

1 Introduction

There are various object recognition methods in images that generally require engineering features for each object category, it is less efficient for datasets that have many categories. Artificial neural network is a computational method inspired by biological neural networks and can be used for pattern recognition in data. Deep learning is part of machine learning that focuses on non-linear information processing (such as images) that has many layers of representation of data with different levels of abstraction to be able to model complex relationships between data, which usually use artificial neural networks with supervised or unsupervised learning methods for pattern analysis, feature extraction and classification [1].

Convolutional neural network is a type of deep feed forward neural network that is easier to train for image datasets and has better generalization capabilities than ordinary artificial neural networks, this type of network is specifically designed to process data in the form of multidimensional arrays such as pixel data in color image which is a two-dimensional array for each color channel [2]. One of the successes of the convolutional neural network in 2012 was that the network was trained in a large-scale image dataset, namely ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC) with a top-5 error rate of 15.3% in the test data [3].

Convolutional neural network capability in generalizing training data (training data) and producing a fairly accurate classification can also be used as a digital image extractor feature that represents the characteristics of the image in the form of arrays or vectors so that each feature extraction performed on each image can be stored and entered into a model that can calculate the similarity value between two images or more using a distance metric to be sorted according to the similarity of content in the image, but this model also has many parameters and design decisions are difficult to determine to be able to get good performance.

Applies a convolutional neural network algorithm to classify images, which uses training and test data on the Caltech 101 dataset, with 5 categories of poultry, namely: Emu, Flamingo, Ibis, Pigeon, and Rooster, besides 3 other categories, namely: Cougar (Cougar Body and Cougar

ICONISTECH-1 2019, July 11-12, Bandung, Indonesia

DOI 10.4108/eai.11-7-2019.2297409

Face), Crocodile (Crocodile and Crocodile Head) and Face (Face and Face Easy), the test results in the test data produced the lowest percentage of 20% in the poultry category where the categories Ebis, Flamingo, Pigeon and Emu cannot be classified properly , then a percentage of 50% in the Cougar, Crocodile, and Face categories, so the overall percentage is between 20% - 50% [4].

Research using convolutional neural network algorithm to recognize faces, the dataset used was The Extended Yale Face Database B in the form of face photos, also applied the dropout regularization method in the convolutional neural network architecture which had the best recognition accuracy of 89.73%, while in the data test of 75.79% [5].

Research uses two methods namely convolutional neural network with support vector machine which is used for human action recognition. The first convolutional neural network is used to extract the spatial and temporal features of video frames and then use the support vector machine to classify each extracted feature. this architecture is trained and evaluated on KTH action recognition datasets and has good performance with an accuracy of 90.34% [6].

Convolutional neural networks to recognize handwriting letters and numbers. The system is built using C # programming language with Visual studio 2010, and with test results on data in the form of single letters and numbers 184 with correct answers 153 and wrong 31, and the results of trials on input words with many letters 191 with the number of correct answers as much as 158 and incorrect answers as many as 33 letters that cannot be recognized correctly [7].

Trains convolutional neural networks using VGGNet architecture with several changes using pretrained networks (transfer learning) and training networks from the beginning using the NUS-WIDE-SCENE dataset. CNN is used to classify images and use them to give labels (multilabel) to an image taken from the results of top-3 or top-5 classifications, where the labels can be used to base a content based image retrieval (CBIR) system, with has top-5 accuracy in the test data of 81.24%. Based on the results of previous studies, most of them used the convolutional neural network as a classifier for various problems. In these studies convolutinoal neural network shows a fairly good classification accuracy performance [8].

2 Research Method

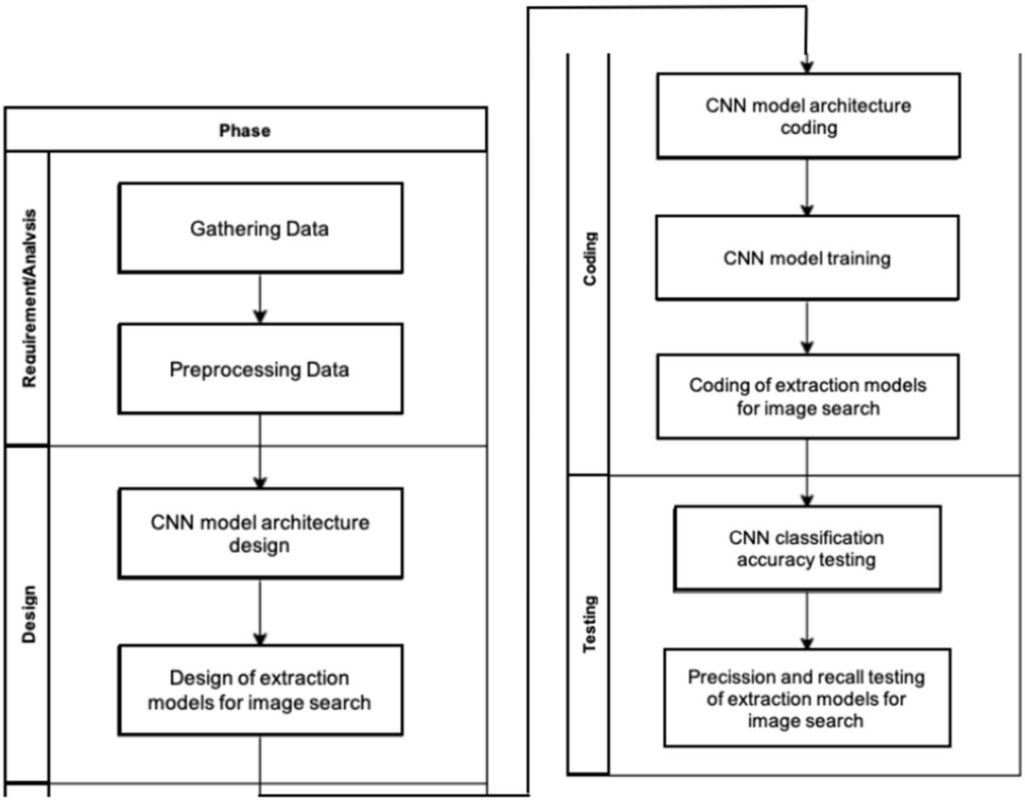
Phase
Requirement/Analysis
Gathering Data
Preprocessing Data
Design
CNN model architecture design
Design of extraction models for image search
Coding
CNN model architecture coding
CNN model training
Coding of extraction models for image search
Testing
CNN classification accuracy testing
Precission and recall testing of extraction models for image search

Fig. 1. Stages of research that will be conducted

2.1 Data Preprocessing

Before the data can be used for training and testing the convolutional neural network model or pre-processing the image will be carried out first. Because the size of the data to be collected must have a different size (resolution) while the input data for the convolutional neural network model requires a fixed size, the image size will be synchronized. The size that will be used is 128x128 pixels with three RGB color channels (Red, Green, Blue).

2.2 Feature Extraction Process

There is a gallery image, the image that will be searched by the search model. then the query image is an image that is entered as a key image to search the gallery image. Gallery images and query images will be taken from the test data. In this image feature classification and extraction processes will be carried out using the convolutional neural network model that has been trained, then the top-1, top-2 classification labels and extracted vector features will be stored in the database.

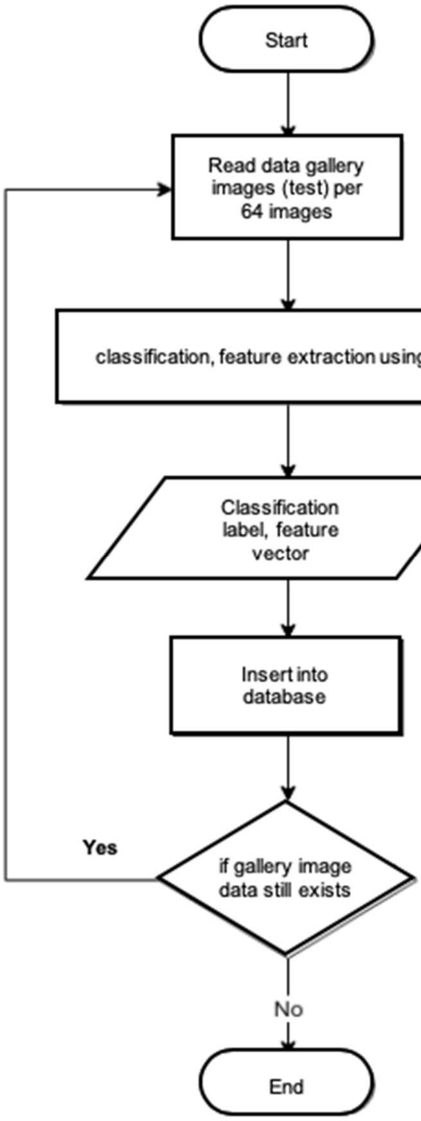
Start
Read data gallery images (test) per 64 images
classification, feature extraction using CNN
Classification label, feature vector
Insert into database
if gallery image data still exists
Yes
No
End

Fig. 2. Image feature extraction flow

2.3 Image Search Design

The process starts from the query image, the read image is then classified and extracted using the convolutional neural network model, then produces the top-1 and top-2 labels and the query vector image features. Next, query the gallery image database using the two labels that have been obtained. The images that are successfully obtained will be sorted using distance metrics, in this study using two methods namely cosine and euclidean distance.

2.4 Image Search Design

The process starts from the query image, the read image is then classified and extracted using the convolutional neural network model, then produces the top-1 and top-2 labels and the query vector image features. Next, query the gallery image database using the two labels that have been obtained. The images that are successfully obtained will be sorted using distance metrics, in this study using two methods namely cosine and euclidean distance.

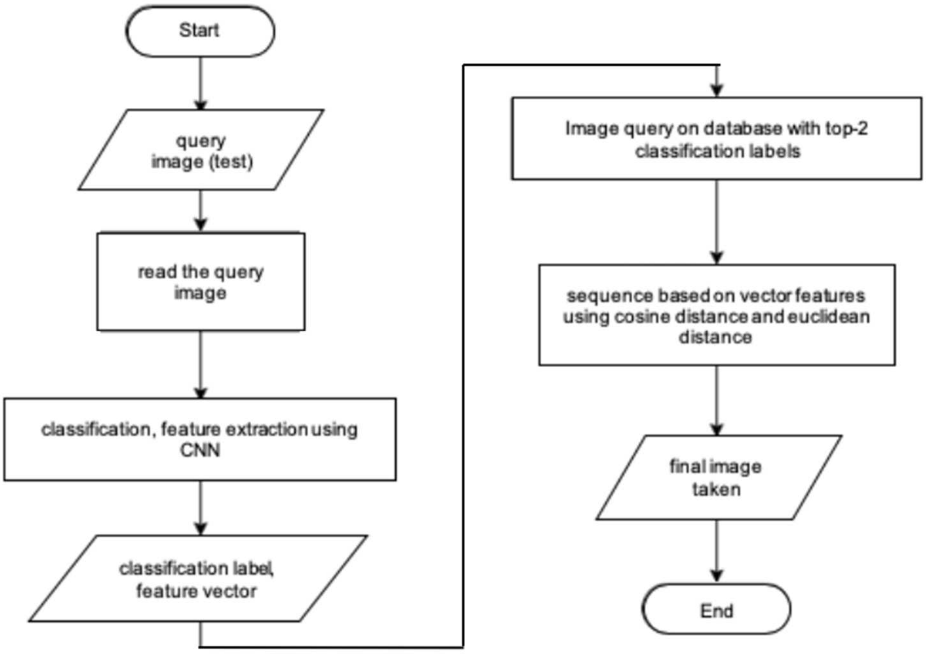
Start
query image (test)
read the query image
classification, feature extraction using CNN
classification label, feature vector
Image query on database with top-2 classification labels
sequence based on vector features using cosine distance and euclidean distance
final image taken
End

Fig. 3. Image search process flow

2.5 Search Testing Process

To determine the performance of the convolutional neural network model as an extractor feature and overall image search, the value of precision and recall will be calculated. The value of precision and recall obtained will be calculated the value of F-measure. The value of precision and recall for a category will be taken from the average value of precision and recall of all images in that category, which can be formulated in equation 3.

$$P^k = \frac{\sum_{i=1}^{m} P_i}{m}$$
$$R^k = \frac{\sum_{i=1}^{m} R_i}{m} \quad (3)$$

Pk = Average results of precision for a category
Rk = Average recall results for a category
Pi = Precision value to - i
Ri = Recall value to - i
m = Amount of data in category
Calculation of precision and recall across categories is taken on average from all precision and recall values in each query can be calculated using equation 4.

$$P^A = \frac{\sum_{i=1}^{n} P_i}{n}$$
$$R^A = \frac{\sum_{i=1}^{n} R_i}{n} \qquad (4)$$

PA = Average results of precision of all data
RA = The results of the average recall of all data
n = Amount of data used as a query (whole)
Precision recall and f-measure tests were carried out on each model experiment, namely 6 convolutional neural network architectural models. The test has 20 different threshold distance metric configuration scenarios, namely cosine distance has 10 threshold with a range of 0.1 - 1.0 and euclidean distance has 10 thresholds with a range of 10.0 - 100.0.

2.6 Equipment

Computer development and design of convolutional neural network models with the Python 3.x development environment (PyCharm IDE).
VPS computers on the Google Cloud Platform (Compute Engine) for training convolutional neural network models with GPU hardware (Graphics Processing Unit).
Deep learning framework Tensorflow (especially on computer development and training.

3 Results and Disscusion

3.1 Data Collection

Data was taken from various sources, namely the shoe category from UT Zappos50K dataset [9], mug, teapot, microwave category from ImageNet dataset, wristwatch, backpack category from blibli.com. The data obtained were 11,372.

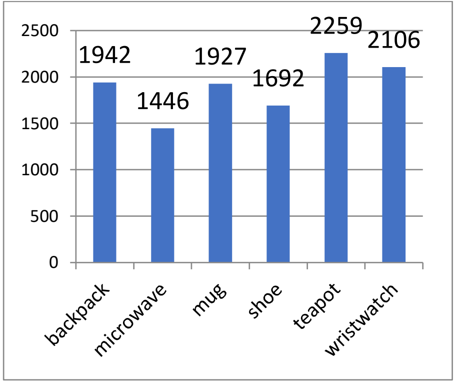

Fig. 4. Number of images per category

Before use the image size is synchronized to 128x128 pixel RGB (Red, Green, Blue). The results of this process are as follows.

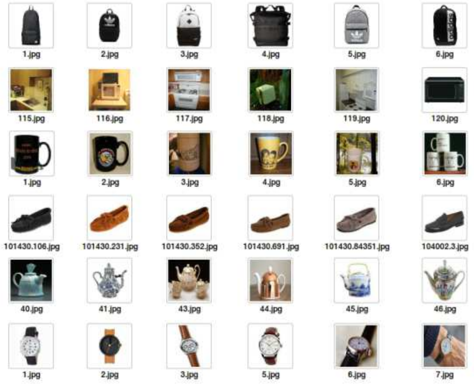

Fig. 5. Image of the same size

3.2 Split Data

After the data is processed, the data is broken down into three parts, namely training data, validation data and test data, data broken down with a ratio of 60% training, 20% validation and 20% test.

Table 1. Results of split data

Category	Training	Validation	Test
backpack	1165	389	388
microwave	868	289	289
mug	1156	386	385
shoe	1015	339	338
teapot	1355	452	452
wristwatch	1264	421	421
Total		11372	

3.3 Model Design Result

There are six experimental models, but the results of designing the best convolutional neural network model are cnnarch1_1_do3 as detailed in Table 2.

Table 2. Details of the order of architectural data flow convolutional neural network cnnarch1_1_do3

Urutan Layer CNN	
Input 128x128x3	
Layer 1	Conv 3x3, 32
	ReLU
Output 128x128x32	
Layer 2	Conv 3x3, 64
	ReLU

	Max Pooling 2x2
Output 64x64x64	
Layer 3	Conv 3x3, 64
	ReLU
	Max Pooling 2x2
Output 32x32x64	
Layer 4	Conv 3x3, 128
	ReLU
	Max Pooling 2x2
Output 16x16x128	
Layer 5	Conv 3x3, 128
	ReLU
	Max Pooling 2x2
Output 8x8x128	
Layer 6	Conv 3x3, 256
	ReLU
	Dropout 0.5
	Max Pooling 2x2
Output 4x4x256	
Flatten	
Output 4096	
Layer FC 1	Fully Connected 512
	ReLU
	Dropout 0.5
Output 512	
Layer FC 2	Fully Connected 1024
	ReLU
	Dropout 0.5
Output 1024	
Output Layer	Fully Connected 6
Output 6 Skor Kategori	

The activation function used is ReLU because it has good performance [3].

3.4 Training Result

Training is conducted on the google vps computer compute engine with 8vCPU hardware specifications, 10GB RAM, NVIDIA Tesla K80, 30GB SSD. The training results are shown in table 3.

Table 3. The results of the training model cnnarch1_1_do3

Epoch	198
Duration	98 minutes

Loss Training	0,2159
Training Accuracy	92,67%
Loss Validation	0,3583
Validation Accuracy	90,3%

3.5 Classification Test Result

Table 4. Test results of test data

Top-1 Test Accuracy	87,72%
Top-2 Test Accuracy	96,08%

In table 4 shows that the top-1 classification accuracy has a pretty good value of 87.72%, and top-2 is 96.08%. The following in table 5 is a breakdown of the percentage of the top-1 test.

Table 5. Details of test data test results

Category	Amount of data	Correct Classification	Top-1 Testing Accuracy (%)
backpack	388	351	90.46
microwave	289	256	88.58
mug	385	254	65.97
shoe	338	336	99.40
teapot	452	411	90.92
wristwatch	421	386	91.68
TOTAL	2273	1994	87.72

3.6 Image Search Test Result

The query image data used is taken from the same test data as the gallery image, the test images are taken one by one to be used as search query images, therefore the search and precision and recall calculations are carried out as many as 2273 times. Testing precision, recall and f-measure were carried out on the best model, with experiments on the last 3 layers of the model, namely fully connected 2 (FC 2), fully connected 1 (FC 1) and flatten layers, with each of the 20 threshold configuration scenarios.

3.7 Comparison of cosine distance and euclidean distance

In Figure 6 shows a comparison of distance metric cosine and euclidean performance on the best model, namely cnnarch1_1_do3 on three feature extraction layers, namely fully connected 2 (FC2), fully connected 1 (FC1) and Flatten.

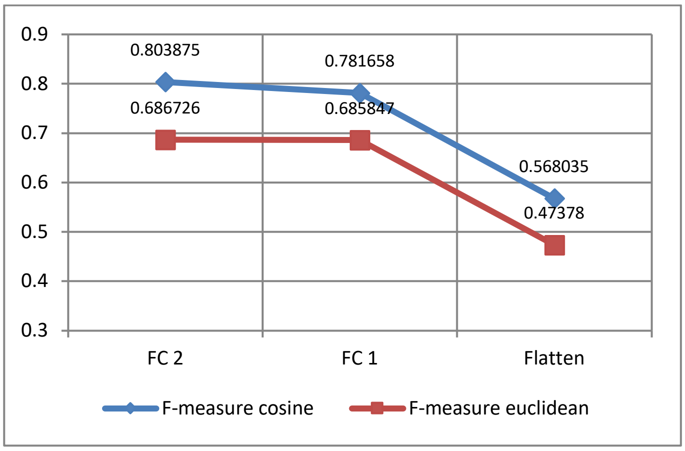

Fig. 6. Comparison of cosine and euclidean performance

Most cosine distance experiments have better performance than euclidean distance, and the best model in Figure 6 also shows that cosine has better performance.

3.8 Comparison of layers for feature extraction

The f-measure score obtained at FC 2 layer (Fully connected 2) is higher than FC layer 1 (Fully connected 1) and flatten layer using cosine distance. As shown in Figure 6 and Figure 7.

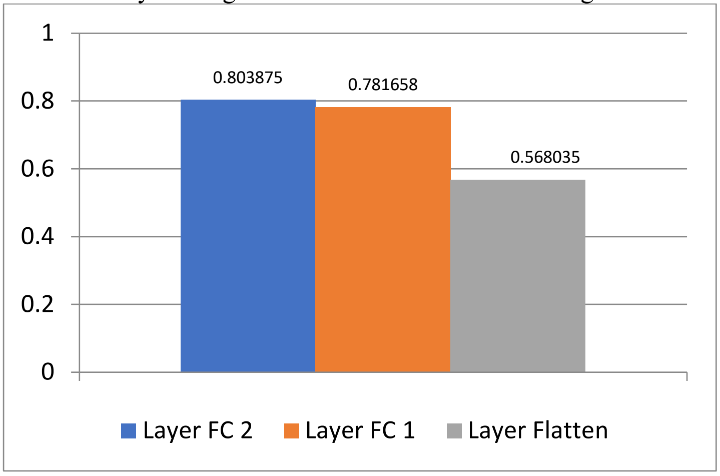

Fig. 7. Comparison of extraction layers

The results of the best configuration obtained are at FC 2 layer (Fully connected 2) using the distance metric Cosine distance with a distance threshold of 0.4 as shown in table 6.

Table 6. The best value for precision and recall

Category	Precision	Recall	F-measure
Mug	0.522593	0.713098	0.603161
Teapot	0.613761	0.896145	0.728547
backpack	0.852994	0.845281	0.84912
Shoe	0.982684	0.993943	0.988281
wristwatch	0.844945	0.841899	0.843419
microwave	0.810846	0.815927	0.813378
All Categories	0.761893	0.850754	0.803875

4 Conclusion

Making a convolutional neural network model using the TensorFlow framework could accelerate the development process because it had a variety of mathematical operations that were ready to use. The number of parameters in the model also affected the performance of the model itself, especially the use of dropouts. Model training was carried out on high-performance computers using the GPU so the training process was fast. The best convolutional neural network model had a top-1 test accuracy of 87.72%, top-2 of 96.08%. The best feature extraction performance at the second last fully connected layer, FC 2, used a cosine distance with a threshold of 0.4 resulting in a fairly good value of 76.18% overall precision, 85.07% recall and an f-measure score of 80.38%.

5 Acknowledgement

This research was supported by Universitas Majalengka. We thank our colleagues from Informatics Department who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

6 References

[1] Deng L and Yu D 2013 Deep learning: Methods and applications Found. Trends Signal Process. 7 197–387

[2] Lecun Y, Bengio Y and Hinton G 2015 Deep learning Nature

[3] Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks Advances in Neural Information Processing Systems

[4] Eka Putra W S 2016 Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101 J. Tek. ITS

[5] Abhirawan H, Jondri and Arifianto A 2017 Pengenalan Wajah Menggunakan Convolutional Neural Networks (CNN) Univ. Telkom 4 4907–16

[6] Latah M 2017 Human action recognition using support vector machines and 3D convolutional neural networks Int. J. Adv. Intell. Informatics

[7] Sam’ani and Qamaruzzaman M H 2017 Pengenalan Huruf Dan Angka Tulisan Tangan Mengunakan Metode Convolution Neural Network (CNN) J. Speed – Sentra Penelit. Eng. dan Edukasi 9 55–64

[8] Dzaky Anwari N and Arifianto A Multilabel Image annotation Menggunakan Metode Convolutional Neural Network

[9] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A C and Fei-Fei L 2015 ImageNet Large Scale Visual Recognition Challenge Int. J. Comput. Vis.