

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/342585186>

Analysis of Architecture Combining Convolutional Neural Network (CNN) and Kernel K-Means Clustering for Lung Cancer Diagnosis

Article in *International Journal on Advanced Science Engineering and Information Technology* · June 2020

DOI: 10.18517/ijaseit.10.3.12113

CITATIONS

26

READS

2,122

5 authors, including:



Zuherman Rustam
University of Indonesia

184 PUBLICATIONS 1,744 CITATIONS

[SEE PROFILE](#)



Sri Hartini
University of Indonesia

36 PUBLICATIONS 196 CITATIONS

[SEE PROFILE](#)



Reyhan Eddy Yunus
University of Indonesia

32 PUBLICATIONS 177 CITATIONS

[SEE PROFILE](#)



Rahmat Hidayat
Politeknik Negeri Padang

46 PUBLICATIONS 423 CITATIONS

[SEE PROFILE](#)

Analysis of Architecture Combining Convolutional Neural Network (CNN) and Kernel K-Means Clustering for Lung Cancer Diagnosis

Zuherman Rustam^{a,*}, Sri Hartini^a, Rivan Y. Pratama^a, Reyhan E. Yunus^b, Rahmat Hidayat^c

^a Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia
E-mail: *rustam@ui.ac.id

^b Department of Radiology, Cipto Mangunkusumo Hospital, Jakarta 10430, Indonesia

^c Department of Information Technology, Politeknik Negeri Padang, 25164, Indonesia
E-mail: rahmat@pnp.ac.id

Abstract— In this paper, we proposed the modified deep learning method that combined Convolutional Neural Network (CNN) and Kernel K-Means clustering for lung cancer diagnosis. The Anti-PD-1 Immunotherapy Lung dataset obtained from The Cancer Imaging Archive was used to evaluate our proposed method. From this dataset, we use 400 Magnetic Resonance Imaging (MRI) images that manually labeled consists of 150 healthy lung images and 250 lung cancer images. As the first step, all the data was examined through the CNN architecture. The flatten neuron of the feature map for every image resulted from the convolutional layers in CNN is gained and passed through the kernel k-means clustering algorithm. This algorithm then used to obtain the centroid of each cluster that determines the prediction class of every data point in the validation set. The performance of our proposed method was evaluated using several k values in k-fold cross-validation. According to our experiments, our proposed method achieved the highest performance measure with 98.85 percent accuracy, 98.32 percent sensitivity, 99.40 percent precision, 99.39 percent specificity, and 98.86 percent F1-Score when using RBF kernel function with $\sigma=0.05$ in 9-fold cross-validation. Those performance improves 1.31% sensitivity, 1.12% accuracy, 1.11% F1-Score, 0.92% specificity, and 0.91% precision compared to when using 5-fold cross-validation. It is even obtained in less than 8 seconds for passing the dataset to the CNN model and 40 ± 0.77 seconds for examined in kernel k-means clustering. Therefore, it was proved that our proposed method has an efficient and promised performance for lung cancer diagnosis from MRI images.

Keywords— artificial intelligence; artificial neural network; deep learning; image classification; kernel function; k-means clustering; lung cancer diagnosis.

I. INTRODUCTION

The high popularity of deep learning was already expected due to the increase in the volume and complexity of data. In the medical applications, Convolutional Neural Networks (CNN) is the most popular among the other commonly used deep learning algorithms, such as recurrent neural networks, deep belief networks, and deep neural networks [1]. This method is widely used when it comes to medical image analysis from a Computerized Tomography (CT) scan or Magnetic Resonance Imaging (MRI) results [2]. González et al. [3] used CNN to examine the type of chronic lung illness prognosis from the CT-scan image. Sun et al. [4] compared the performance of CNN, Deep Belief Networks (DBN), Stacked Denoising Autoencoder (SDAE) using CT-scan image to diagnose lung cancer and concluded DBN as the method that has the highest accuracy (0.8119). Anirudh

et al. [5] also used CNN in CT-scan images for lung nodule detection when there is weak label information. Meanwhile, Causey et al. [6] used CNN to classify lung cancer nodule malignancy, and his approach achieved 99% accuracy. The combination of CNN and the other methods seems to be developed in the last few years. The U-Net with 3D CNN was introduced by Chon et al. [7], which deliver 70% accuracy. Winkels and Cohen [8] also tried to use 3D rotation group convolutions in CNN to achieve a Free-Response Operating Characteristic (FROC) score close to the CNN with fewer data. Meanwhile, Zhang et al. [9] conducted the promising Mask Region-Based Convolutional Neural Network (Mask R-CNN) architecture, which has 90% precision, 100% recall, and 95% F1-score.

Moreover, the combination of the CNN model architecture could also happen with the robust classifier in machine learning. For instance, CNN has been combined with Support Vector Machines (SVM), which has proved to

gain higher accuracy than original CNN, by Leng et al. [10] for image classification, Niu et al. [11] for recognizing handwritten digits, and Elleuch et al. [12] for Arabic Handwritten Recognition. There are machine learning methods that commonly used to classify data such as Random Forest (RF) and SVM. Those methods have been used in any field, including medical and financial. For example, RF has been used for predicting osteoarthritis disease by Apriliani et al. [13] and prostate cancer by Huljanah et al. [14]. Meanwhile, SVM has been used in classifying policyholders satisfactorily in automobile insurance by Rustam et al. [15] and schizophrenia data by Rampisela et al. [16]. It was also used for detecting cancer data by Nadira et al. [17], and also has been used for face recognition by Rustam et al. [18]. Both of those methods frequently have accuracy higher than 90 percent.

However, instead of using a classifier in the machine learning algorithm for the classification task, the clustering method seems compelling to use. In the previous research, many clustering methods have already been used for classifying. Among them, there was kernel spherical k-means used by Arfiani et al. [19] for distinguishing acute and chronic sinusitis. There was also fuzzy kernel k-medoids used by Rustam and Talita [20] for detecting anomaly problems. In terms of their performance, clustering methods have the same chance to develop to do the classification task.

Therefore, in this research, we proposed combining CNN architecture with the kernel k-means clustering in this paper. Kernel k-means clustering is chosen due to its simplicity in implementation and its ability because of the use of kernel. According to the experiments done by Guérin and Boots [21], we can conclude that our proposed method is feasible because CNNs can be used as feature extractors for many applications, and clustering is no exception.

II. MATERIALS AND METHODS

A. Dataset

The dataset used in this research is the Anti-PD-1 Immunotherapy Lung dataset obtained from The Cancer Imaging Archive [22]. This collection includes 46 lung cases treated with anti-PD1 immunotherapy in 2016, each with pre-treatment and most with one imaging follow-up timepoint. However, we only use 400 MRI images that manually labeled consists of 150 healthy lung images and 250 lung cancer images. The size of all the images used in this paper is 152 x 152 x 1 pixels.

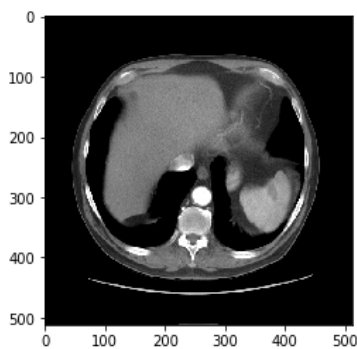


Fig. 1 The sample of the lung cancer MRI image

For the example of lung cancer image (see Fig. 1), the existence of the cancer is indicated by the presence of blood inside the lung. Meanwhile, for the example of a healthy lung image (see Fig. 2), the dark gray area inside the area of ribs only shows the existence of the heart, and there is no presence of blood.

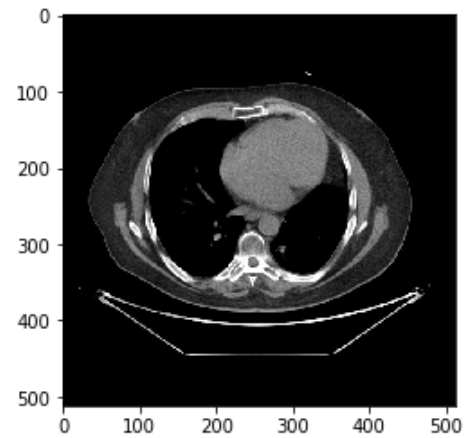


Fig. 2 The sample of healthy lung MRI image

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is one of the deep feed-forward artificial neural network architectures that is frequently used in computer vision problems such as image classification. The difference between CNN and multilayer perceptron (MLP) network is its usage of convolutional layers, pooling, and non-linearities such as tanh, sigmoid, and ReLU [23].

In this paper, we used the architecture combining CNN and Kernel K-Means clustering with the RBF kernel function with several values of σ . The performance of our proposed method was evaluated using several k values in k-fold cross-validation. As the first step, all the data was examined through the CNN architecture (see Fig. 3) that consists of the convolutional layer, activation function, pooling layer, and normalization layer. In this paper, Keras, as the python deep learning library, is used to build the model architecture.

Consider an input grayscale image with a size of 152 x 152 x 1 pixels. This input image will be processed through a convolutional layer with the Rectified Linear Unit (ReLU) activation function, maximum pooling layer, and normalization layer, respectively, before finally be applied dropout and make the output flat.

Each convolutional neural network consists of a different number of convolution layers depending on the network requirements [24]. In this paper, five convolutional layers with the same size 3 x 3 are used. In the first convolutional layer, we use 32 filters and then 64 filters in the second convolutional layer. In the middle, or the third convolutional layer, 96 filters is used. The number of filters then back to 64 and 32 filters in the fourth and last convolutional layer, respectively. The amount of filters in the convolutional layer is chosen wisely considering the output size of flatten layer. The output is managed so that it is not huge for being an input in kernel k-means clustering.

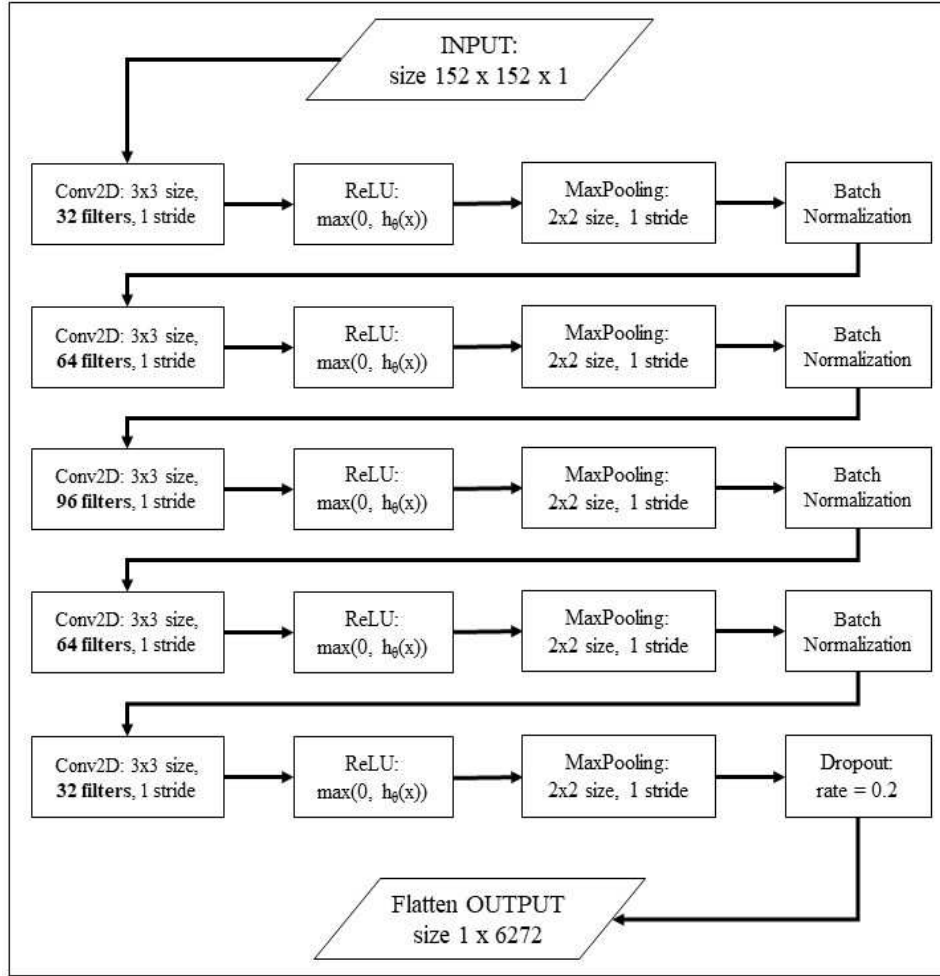


Fig. 3 The CNN architecture used in this paper

In the convolutional layers, we use a convolution process through the utilization of filters inside, which has similar size as that of an image in general, including the length and height (pixels), as well as width [25], as illustrated in Fig. 4 [26].

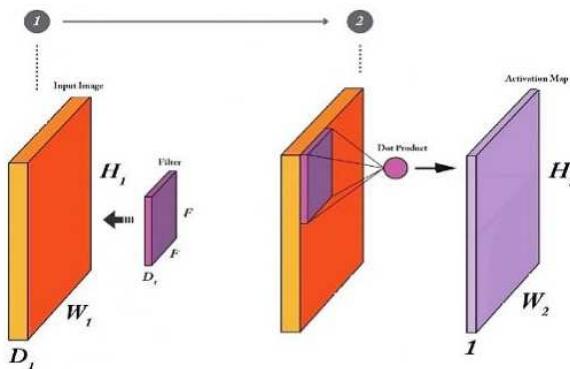


Fig. 4 Illustration of convolutional layer

This layer constructs a convolution kernel that is convolved with the layer input to produce a tensor of outputs through the dot product computation. Convolution operation extracts useful features from locally correlated

data points. The output of the convolutional kernels is assigned to the non-linear processing unit (activation function), which not only helps in learning abstractions but also embeds non-linearity in the feature space [27].

Consequently, an activation function is used for familiarizing non-linearities in the computation so that the model does not only learn linear mappings. As the activation function, ReLU is used to faster computations compared to other activation functions such as tanh and sigmoid functions. This activation function is also the most widely used activation function for deep learning because of its ability to deliver better performance and generalization in deep learning compared to tanh and sigmoid activation function [28]. ReLU is a linear and numerically less complicated function that retains the positive values of the input, while the negative values are turned to zero [29]. Simply thresholding matrix values implement ReLU at zero (see Eq. 1)

$$f(h_0(x)) = \max(0, h_0(x)) \quad (1)$$

The pooling layer is then used to decrease the resolution of the image in order to decrease the number of parameters that also cause a decrease in the computational burden [30]. the pooling process commonly used includes: (1) max

pooling, where the use of 2x2 with stride 2 leads to the maximum value of 2x2 pixel area selected for each shift, (2) average pooling, which chooses its mean value. In this paper, max-pooling was used over the average pooling because it is commonly used in CNN architecture.

The normalization layer was then used after that due to the purpose of preventing model divergence [31]. The normalization layer used to normalize the activations of the previous layer at each batch. For example, it applied a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

As the last step in CNN model architecture, the dropout is used so that the network can generalize better, and the influence of an individual neuron on the output produced is reduced [32]. We use dropout = 0.2 in this paper; therefore, 80 percent of the information on every image is kept. As a result, we can see the output image after passing the last convolutional layer in Fig. 5 for the lung cancer image and Fig. 6 for the healthy lung image.

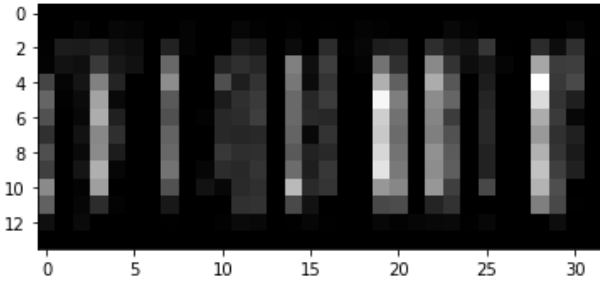


Fig. 5 The result of the last convolutional layer for the image in Figure 1

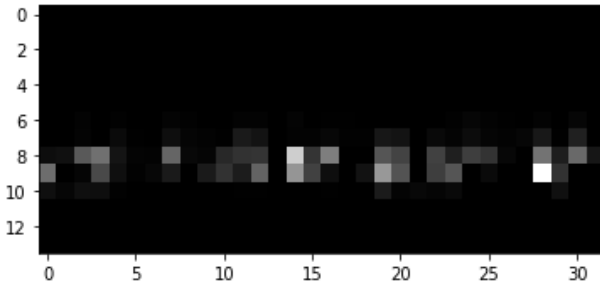


Fig. 6 The result of the last convolutional layer for the image in Figure 2

The neuron resulted from the dropout layer is then flattened and form a vector with length 6272 for every image. The flatten neuron resulted by CNN represents the data image with a smaller size [33]. These outputs were then compiled to matrix data; therefore, we then have matrix 400 x 6272, which contains the flatten output of lung cancer image and healthy lung image. After getting the matrix, we broke down this matrix into two matrices. One matrix 250 x 6272 for the image labeled lung cancer (class 1) and one matrix 150 x 6272 for the image labeled healthy lung (class 0).

Because we used k-fold cross-validation, one fold was used to find the centroid of each cluster while the k-1 folds used as the validation set. The prediction class of the dataset in k-1 folds were determined according to its nearest centroid. If the data point is closer to the centroid of class 1, then the prediction class of this data point is 1. Meanwhile, if it is nearer to the centroid of the class 0, then the prediction class of it is 0.

C. Kernel K-Means Clustering

To find the centroid of each cluster, we followed the algorithm of kernel k-means clustering (see Fig. 7).

Input:
 $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, C (the number of cluster),
 ϵ (epsilon), T (the maximum number of iterations).

Output:
 $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_C\}$,
 $U = [r_{nc}]$ where $n = 1, 2, \dots, N$ and $c = 1, 2, \dots, C$.

1. Initialization: $V^0 = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_C\}$,
2. Update membership of the data point \mathbf{x}_i in j^{th} -cluster.
3. If $c = \text{argmin}_j (K(\mathbf{x}_n, \mathbf{x}_n) - 2K(\mathbf{x}_n, \mu_c) + K(\mu_c, \mu_c))$, then $r_{nc} = 1$. Otherwise, $r_{nc} = 0$.
4. Update cluster center V^t using:
 $\mu_c = (\sum_n r_{nc} \mathbf{x}_n) / \sum_n r_{nc}$, $c = 1, 2, \dots, C$
5. Check the stop criteria. If $\|V^{(t-1)} - V^{(t)}\| < \epsilon$ or $T = t$, then the iteration stops. Otherwise, $t = t+1$ and go back to step 2;
6. End.

Fig. 7 The algorithm of Kernel K-Means clustering [34]

K-Means (KM) clustering based on kernel came from K-Means clustering that was introduced by Lloyd [35] in 1982 as the non-probabilistic technique that used to cluster the data. Consider dataset $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, which has N data points where \mathbf{x} is the D -dimensional data point. The centroid of each cluster was defined as μ_c where $c = 1, 2, \dots, C$, while $r_{nc} \in \{0, 1\}$ where $c = 1, 2, \dots, C$ was defined as the indicator whether the data points are in the cluster (denoted as 1) or not (denoted as 0).

The goal of KM clustering is to minimize its objective function (see Eq. 2) so that every data point is in the cluster where the centroid is nearest [36].

$$J_{KM} = \sum_n \sum_c r_{nc} \|\mathbf{x}_n - \mu_c\|^2 \quad (2)$$

Assumed the mapping function ϕ which mapped the input data \mathbf{x} to the higher dimensional feature space $\phi(\mathbf{x})$. Because now we work in this feature space, the objective function in Eq. 2 becomes as shown in Eq. 3.

$$J_{KM} = \sum_n \sum_c r_{nc} \|\phi(\mathbf{x}_n) - \phi(\mu_c)\|^2 \quad (3)$$

Considering the computational cost of the distance of two points in the higher dimensional feature space, Vapnik [37] defined the kernel function (see Eq. 4) for every $\mathbf{x} \in \mathbb{R}^n$ as the dot product of the mapped result of \mathbf{x} in the feature space.

$$K(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}))^T \phi(\mathbf{y}) \quad (4)$$

According to Eq. 4, the distance between $\phi(\mathbf{x}_n)$ and $\phi(\mu_c)$ can be computed, as shown in Eq. 5.

$$\|\phi(\mathbf{x}_n) - \phi(\mu_c)\|^2 = K(\mathbf{x}_n, \mathbf{x}_n) - 2K(\mathbf{x}_n, \mu_c) + K(\mu_c, \mu_c) \quad (5)$$

Therefore, the objective function of Kernel K-Means clustering becomes the Eq. 6 with the same goal as the KM clustering [34].

$$J_{KKM} = \sum_n \sum_c r_{nc} (K(\mathbf{x}_n, \mathbf{x}_n) - 2 K(\mathbf{x}_n, \boldsymbol{\mu}_c) + K(\boldsymbol{\mu}_c, \boldsymbol{\mu}_c)) \quad (6)$$

Also, $K(\mathbf{x}, \mathbf{y})$ is defined as the kernel function [37] of two data points. In this paper, the Radial Basis Function (RBF) kernel function [38] was used using the several values of σ in these experiments, where the formula shown in Eq. 7.

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / (2\sigma^2)) \quad (7)$$

D. Our Proposed Method

Unlike previous researches in image classification that combined CNN to other deep learning techniques such as Recurrent Neural Network (RNN) by Yin et al. [39] and Long-Short Term Memory (LSTM) by Aditi et al. [40], we proposed the method that combined widely-used CNN with one of the well-known machine learning methods. However, it is not Support Vector Machines (SVM) like what Sugg [41] did in his thesis, or Copur et al. [42] did in their paper, but we combine CNN with KM clustering based on RBF kernel instead.

First, using 5-fold cross-validation, we conducted the experiments to find the optimal RBF parameter σ . After that, the value of σ is then used to several k in k-fold cross-validation. The flowchart of our experiment is illustrated in Figure 8.

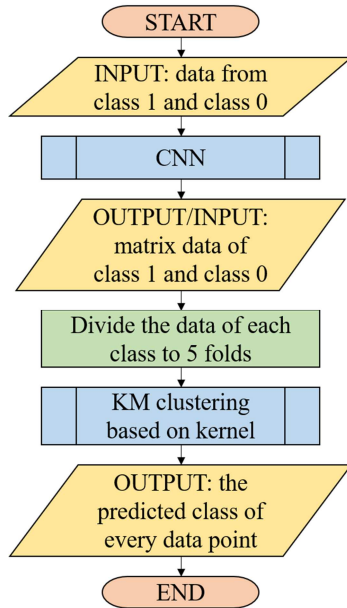


Fig. 8 The flowchart of CNN + K-Means Clustering based on RBF kernel that we used in this research

As the input of CNN, we used all of 400 labeled images: 1 for lung cancer images and 0 for healthy lung images. The images are resized to the same size 152 x 152 pixels. This input is then passed to the bunch of CNN layers, as described in Figure 3. As a result, every image 152 x 152 x 1 pixel became a vector with length 6272. Therefore, we now have matrix 334 x 6272, where the row index is indicated the image that we observed and the column index is indicated the feature map resulted by CNN.

The next step is then dividing the matrix data according to its label, whether it is included in class 1 (lung cancer) or class 0 (healthy lung). In this case, we have two matrices

data with size 150 x 6272 for the healthy lung and 250 x 6272 for lung cancer class.

Those matrices were then used in k-fold cross-validation for evaluating KM clustering based on kernel algorithm. For example, when we used 5-fold cross-validation, the data is divided into five folds for each class. Therefore, we get the number of points in every fold as shown in Table I.

TABLE I
THE NUMBER OF DATA IN EVERY 5 FOLDS OF LUNG CANCER DATASET

Fold	The number of lung cancer data points	The number of healthy lung data points
1	50	30
2	50	30
3	50	30
4	50	30
5	50	30
Total	250	150

The k-fold cross-validation using KM clustering based on kernel might be different from the usual used in supervised learning method in machine learning. In this KM clustering based on kernel, a fold was used to obtain the centroids of the clusters according to the algorithm in Figure 7, while the rest k-1 folds were used to evaluate the method by determining the class of every data point according to its nearest centroid. If the data point was nearer to the centroid of class 1, then the predicted class for this data point is 1. Meanwhile, if the data point was nearer to the centroid of class 0, then the predicted class for this data point is 0.

The performance of our proposed method is examined using the confusion matrix [43]. In the confusion matrix, we can obtain the number of True Positives (TP), True Negative (TN), False Positive (FP), False Negative (FN) that resulted from our model. TP represents the number of lung cancer images that correctly diagnosed as lung cancer, and TN represents the number of healthy lung images that correctly diagnosed as a healthy lung. Meanwhile, the wrong diagnosis is shown by FN and FP. When the actual class is lung cancer, but it was diagnosed as a healthy one, then it counts as FN. On the other side, when the healthy one is diagnosed has lung cancer, it counts as False Positive. The confusion matrix is shown in Table 2.

TABLE II
CONFUSION MATRIX FOR LUNG CANCER DATASET

Confusion Matrix		Predicted Class	
		Lung Cancer	Healthy
Actual Class	Lung Cancer	TP	FN
	Healthy	FP	TN

The performance of our proposed method is examined using the accuracy, sensitivity, precision, specificity, and F1-Score, which the formulas are shown in Eq. (8)-(12) below [44].

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN) \quad (8)$$

$$\text{Sensitivity} = TP / (TP + FN) \quad (9)$$

$$\text{Precision} = TP / (TP + FP) \quad (10)$$

$$\text{Specificity} = TN / (TN + FP) \quad (11)$$

$$\text{F1-Score} = (2 \times \text{Sensitivity} \times \text{Precision}) / (\text{Sensitivity} + \text{Precision}) \quad (12)$$

According to Kotu and Deshpande [45] revealed that the ability of the classifier to select all necessary cases to select and reject all necessary cases to reject is defined as accuracy. The ability of a classifier to select all the necessary cases to select is defined as sensitivity. In the meantime, the proportion of all relevant cases found is defined as precision. Besides, the ability of a classifier to reject all the necessary cases to reject is defined as specificity.

As we noticed in the beginning, we have 150 healthy lung images and 250 lung cancer images, which is an unbalanced data. Therefore, the balanced accuracy or F1-Score is used to confirm that accuracy is crosschecked by balanced data. It is defined as the mean of arithmetic on the class retracting accuracy. It represents the accuracy attained on positive and negative instances, correspondingly.

III. RESULTS AND DISCUSSION

The performance of our proposed method first evaluated using several RBF kernel parameter σ (see Table 3). Several values of σ were chosen from range 10^{-8} to 10^4 . From this table, we can see that their performance varies, but always delivered score above 90 percent in accuracy, sensitivity, precision, specificity, or even F1-Score. Moreover, it was obtained that $\sigma = 0.05$ delivered the highest performance measure among the others. With using $\sigma = 0.05$, we can classify the lung cancer with 97.75% accuracy, 97.04% sensitivity, 98.50% precision, 98.48% specificity, and 97.77% F1-Score.

TABLE III
THE PERFORMANCE OF 5-FOLD CROSS-VALIDATION OF CNN + KERNEL K-MEANS CLUSTERING USING RBF KERNEL FUNCTION FOR LUNG CANCER DIAGNOSIS

σ	Accuracy	Sensitivity	Precision	Specificity	F1-Score
10^{-8}	96.42	96.96	95.83	95.88	96.39
10^{-4}	96.75	96.98	96.50	96.52	96.74
10^{-3}	96.50	94.71	98.50	98.44	96.57
5×10^{-2}	97.75	97.04	98.50	98.48	97.77
10^{-1}	95.50	96.91	94.00	94.17	95.43
1	96.00	96.94	95.00	95.10	95.96
10	95.95	95.45	96.50	96.46	95.97
10^2	95.45	96.33	94.50	94.60	95.41
10^3	94.90	95.35	94.40	94.46	94.87
10^4	94.35	94.31	94.40	94.39	94.35

Furthermore, we then evaluated several values of k in k-fold cross-validation such as k = 5, 7, 9, and 11 to see how the number of folds influences the performance of this method. As the results, we can see in Table 4 that the performance of our proposed method was excellent in all of selected k value; however, it was optimal when using 9-fold cross-validation with 98.85% accuracy, 98.32% sensitivity, 99.40% precision, 99.39% specificity, and 98.86% F1-Score. These scores improved compared to the previous results in Table 2. There are improvements in any aspect of performance measurement. The biggest one is 1.31% improvement in sensitivity, followed by 1.12% improvement in accuracy, and 1.11% in F1-Score. Meanwhile, the precision improves 0.91% and there is 0.92% improvement in specificity.

TABLE IV
THE PERFORMANCE OF CNN + KERNEL K-MEANS CLUSTERING USING RBF KERNEL FUNCTION WITH SIGMA = 0.05 FOR LUNG CANCER DIAGNOSIS

k	Accuracy	Sensitivity	Precision	Specificity	F1-Score
5	97.75	97.04	98.50	98.48	97.77
7	96.35	94.52	98.40	98.33	96.42
9	98.85	98.32	99.40	99.39	98.86
11	97.85	98.28	97.40	97.42	97.84

Therefore, according to Table 3 and Table 4, we obtained the best results when 9-fold cross-validation and RBF kernel parameter $\sigma = 0.05$ is used. For the time computational time that our proposed method needs, it takes less than 8 seconds for passing the dataset to the CNN model and 40 ± 0.77 seconds on average for running the k-fold cross-validation on KM clustering based on RBF kernel. Hence, it shows that our proposed method is not only accurate but also more efficient in time.

IV. CONCLUSION

We analyzed the performance of combined Convolutional Neural Network (CNN) and Kernel K-Means clustering for lung cancer diagnosis. The Anti-PD-1 Immunotherapy Lung dataset obtained from The Cancer Imaging Archive is used. We passed every image in this dataset through the convolutional layers on CNN. After that, we used the compilation of the flatten neuron of the feature map in the form of a matrix to the kernel k-means clustering algorithm for obtaining the centroid of each cluster that determines the prediction class of every data point in the validation set. According to our experiments, our proposed method achieved the best performance with 98.85 percent accuracy, 98.32 percent sensitivity, 99.40 percent precision, 99.39 percent specificity, and 98.86 percent F1-Score when using the RBF kernel function with sigma=0.05 in 9-fold cross-validation. It is even obtained in 48 ± 0.77 seconds. This result proves that our proposed method has the promised accuracy for lung cancer detection from MRI images that even more efficient in time than the usual CNN.

ACKNOWLEDGMENT

The authors were grateful to all the reviewers included in the improvement of this article. This research was financially supported by Universitas Indonesia with PUTI Q2 2020 research grant scheme (ID number NKB-1646/UN2.RST/HKP.05.00/2020).

REFERENCES

- [1] F. Jiang *et al.*, "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4, p. 230, 2017, doi: 10.1136/svn-2017-000101.
- [2] M. Bakator and D. Radosav, "Deep Learning and Medical Diagnosis: A Review of Literature," *Multimodal Technologies Interact*, vol. 2(3), no. 47, 2018, doi: 10.3390/mti2030047.
- [3] G. González *et al.*, "Disease Staging and Prognosis in Smokers Using Deep Learning in Chest Computed Tomography," (in eng), *Am J Respir Crit Care Med*, vol. 197, no. 2, pp. 193-203, Jan 15 2018, doi: 10.1164/rccm.201705-0860OC.
- [4] W. Sun, B. Zheng, and W. Qian, *Computer aided lung cancer diagnosis with deep learning algorithms* (SPIE Medical Imaging). SPIE, 2016.

- [5] R. Anirudh, J. Thiagarajan, T. Bremer, and H. Kim, *Lung nodule detection using 3D convolutional neural networks trained on weakly labeled data* (SPIE Medical Imaging). SPIE, 2016.
- [6] J. L. Causey *et al.*, "Highly accurate model for prediction of lung nodule malignancy with CT scans," *Scientific Reports*, vol. 8, no. 1, p. 9286, 2018/06/18 2018, doi: 10.1038/s41598-018-27569-w.
- [7] A. Chon and N. Balachandar, "Deep Convolutional Neural Networks for Lung Cancer Detection," *Conference Proceedings* 2017.
- [8] M. Winkels and T. S. Cohen, "3D G-CNNs for Pulmonary Nodule Detection," *arXiv e-prints*, p. arXiv:1804.04656. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2018arXiv180404656W>
- [9] R. Zhang, C. Cheng, X. Zhao, and X. Li, "Multiscale Mask R-CNN-Based Lung Tumor Detection Using PET Imaging," (in eng), *Mol Imaging*, vol. 18, p. 1536012119863531, Jan-Dec 2019, doi: 10.1177/1536012119863531.
- [10] J. Leng, T. Li, G. Bai, Q. Dong, and H. Dong, "Cube-CNN-SVM: A Novel Hyperspectral Image Classification Method," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 6-8 Nov. 2016 2016, pp. 1027-1034, doi: 10.1109/ICTAI.2016.0158.
- [11] X. X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318-1325, 2012/04/01/ 2012, doi: 10.1016/j.patcog.2011.09.021.
- [12] M. Elleuch, R. Maalej, and M. Kherallah, "A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition," *Procedia Computer Science*, vol. 80, pp. 1712-1723, 2016/01/01/ 2016, doi: 10.1016/j.procs.2016.05.512.
- [13] U. Aprilliani and Z. Rustam, "Osteoarthritis Disease Prediction Based on Random Forest," in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 27-28 Oct. 2018 2018, pp. 237-240, doi: 10.1109/ICACSIS.2018.8618166.
- [14] M. Huljanah, Z. Rustam, S. Utama, and T. Siswantining, "Feature Selection using Random Forest Classifier for Predicting Prostate Cancer," *IOP Conference Series: Materials Science and Engineering*, vol. 546, p. 052031, 2019/06/26 2019, doi: 10.1088/1757-899x/546/5/052031.
- [15] Z. Rustam and N. P. A. Audia Ariantari, "Support Vector Machines for Classifying Policyholders Satisfactorily in Automobile Insurance," *Journal of Physics: Conference Series*, vol. 1028, p. 012005, 2018/06 2018, doi: 10.1088/1742-6596/1028/1/012005.
- [16] T. V. Rampisela and Z. Rustam, "Classification of Schizophrenia Data Using Support Vector Machine (SVM)," *Journal of Physics: Conference Series*, vol. 1108, p. 012044, 2018/11 2018, doi: 10.1088/1742-6596/1108/1/012044.
- [17] T. Nadira and Z. Rustam, "Classification of cancer data using support vector machines with features selection method based on global artificial bee colony," *AIP Conference Proceedings*, vol. 2023, no. 1, p. 020205, 2018, doi: 10.1063/1.5064202.
- [18] Z. Rustam and R. Faradina, "Face Recognition to Identify Look-Alike Faces using Support Vector Machine," *Journal of Physics: Conference Series*, vol. 1108, p. 012071, 2018/11 2018, doi: 10.1088/1742-6596/1108/1/012071.
- [19] Arfiani, Z. Rustam, J. Pandelaki, and A. Siahaan, "Kernel Spherical K-Means and Support Vector Machine for Acute Sinusitis Classification," *IOP Conference Series: Materials Science and Engineering*, vol. 546, p. 052011, 2019/06/26 2019, doi: 10.1088/1757-899x/546/5/052011.
- [20] Z. Rustam and A. S. Talita, "Fuzzy Kernel k-Medoids algorithm for anomaly detection problems," *AIP Conference Proceedings*, vol. 1862, no. 1, p. 030154, 2017, doi: 10.1063/1.4991258.
- [21] J. Guérin and B. Boots, "Improving Image Clustering With Multiple Pretrained CNN Feature Extractors," *arXiv e-prints*, p. arXiv:1807.07760. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2018arXiv180707760G>
- [22] P. Madhavi, S. Patel, and A. S. Tsao, "Data from Anti-PD-1 Immunotherapy Lung [Data set]," *The Cancer Imaging Archive*, 2019, doi: 10.7937/tcia.2019.zjjwb9ip.
- [23] A. F. Agarap, "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification," *arXiv e-prints*, p. arXiv:1712.03541, 2017. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2017arXiv171203541A>.
- [24] T. Gorach, "Deep Convolutional Neural Networks- A Review," *International Research Journal of Engineering and Technology*, vol. 5, no. 7, 2018.
- [25] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279-2301, 2002/10/01/ 2002, doi: 10.1016/S0031-3203(01)00178-9.
- [26] Z. Rustam, R. Yuda, H. Alatas, and C. Aroef, "Pulmonary rontgen classification to detect pneumonia disease using convolutional neural networks," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, p. 1522, 06/01 2020, doi: 10.12928/telkomnika.v18i3.14839.
- [27] A. Khan, A. Sohail, U. Zahoor, and A. Saeed Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *arXiv e-prints*, p. arXiv:1901.06032. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2019arXiv190106032K>
- [28] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *arXiv e-prints*, p. arXiv:1811.03378. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2018arXiv181103378N>
- [29] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Computer Science*, vol. 131, pp. 977-984, 2018/01/01/ 2018, doi: 10.1016/j.procs.2018.04.239.
- [30] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354-377, 2018/05/01/ 2018, doi: 10.1016/j.patcog.2017.10.013.
- [31] F. Schilling, "The effect of batch normalization on deep convolutional neural networks," *Degree Project in Computer Science and Engineering*, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.
- [32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [33] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611-629, 2018/08/01 2018, doi: 10.1007/s13244-018-0639-9.
- [34] M. Welling, "Kernel k-means and spectral clustering," 2013.
- [35] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129-137, 1982, doi: 10.1109/TIT.1982.1056489.
- [36] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.
- [37] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [38] L. Liu, B. Shen, and X. Wang, "Research on Kernel Function of Support Vector Machine," in *Advanced Technologies, Embedded and Multimedia for Human-centric Computing. Lecture Notes in Electrical Engineering*, vol. 260, Y. M. Huang, H. C. Chao, D. J. Deng, and J. Park Eds.: Springer, Dordrecht, 2013.
- [39] Q. Yin, R. Zhang, and X. Shao, "CNN and RNN mixed model for image classification," *MATEC Web Conf.*, vol. 277, p. 02001, 2019, doi: 10.1051/mateconf/201927702001.
- [40] Aditi, M. Nagda, and P. Eswaran, "Image Classification using a Hybrid LSTM-CNN Deep Neural Network," vol. 8, 10/19 2019, doi: 10.35940/ijeat.F8602.088619.
- [41] B. Sugg, "Convolutional support vector machines for image classification," *Master of Science, Departement of Computer Science, University of Exeter, Exeter, England*, 2018.
- [42] M. Copur, B. M. Ozyildirim, and T. Ibrici, "Image Classification of Aerial Images Using CNN-SVM," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 4-6 Oct. 2018 2018, pp. 1-6, doi: 10.1109/ASYU.2018.8554008.
- [43] K. M. Ting, "Confusion Matrix," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb Eds. Boston, MA: Springer US, 2017, pp. 260-260.
- [44] P. Flach, "Performance Evaluation in Machine Learning: The Good, the Bad, the Ugly, and the Way Forward," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9808-9814, 07/17 2019, doi: 10.1609/aaai.v33i01.33019808.
- [45] V. Kotu and B. Deshpande, *Data Science*, 2 ed. Cambridge: Morgan Kaufmann, 2019.