DAFTAR PUSTAKA

Abhirawa, H., Jondri, M. S., & Arifianto, A. (2017). Pengenalan Wajah Menggunakan *Convolutional Neural Network* Face Recognition Using *Convolutional Neural Network*. *E-Proceeding of Engineering*, *4*, 4907–4916.

Afif, M., Fawwaz, A., Ramadhani, K. N., & Sthevanie, F. (2020). Klasifikasi Ras pada Kucing menggunakan Algoritma *Convolutional Neural Network* (CNN). *EProceedings of Engineering*, *8*(1).

Akçay, S., Kundegorski, M. E., Devereux, M., & Breckon, T. P. (2016). Transfer Learning Using *Convolutional Neural Network*s for Object *Classification* Within X-RAY Baggage Security Imagery. *In 2016 IEEE International Conference on Image Processing (ICIP)*, 1057–1061.

Bayat, O., Aljawarneh, S., Carlak, H. F., International Association of Researchers, Institute of Electrical and Electronics Engineers, & Akdeniz Üniversitesi. (2017). Understanding of a *Convolutional Neural Network*. *Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017)*, 21–23.

Chen, W., Sun, Q., Wang, J., Dong, J. J., & Xu, C. (2018). A Novel Model Based on AdaBoost and Deep CNN for Vehicle *Classification*. *IEEE Access*, *6*, 60445–60455. https://doi.org/10.1109/ACCESS.2018.2875525

Dutt, A., & Dutt, A. (2017). Handwritten Digit Recognition Using *Deep Learning*. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *6*(7), 990–997.

Fu'adah, Y. N., Pratiwi, N. C., Pramudito, M. A., & Ibrahim, N. (2020). *Convolutional Neural Network* (CNN) for Automatic Skin Cancer *Classification* System. *IOP Conference Series: Materials Science and Engineering*, *982*(1). https://doi.org/10.1088/1757-899X/982/1/012005

Hanin, M. A., Patmasari, R., Yunendah, R., & Fu'adah, N. (2021). Sistem Klasifikasi Penyakit Kulit Menggunakan *Convolutional Neural Network* (CNN) Skin DIsease *Classification* System Using *Convolutional Neural Network* (CNN). *EProceedings of Engineering*, , *8*(1), 273–281.

Hendaria, M. P., Asmarajaya, A., & Maliawan, S. (2015). Kanker Kulit. *Fakultas Kedokteran Universitas Udayana.*, 1–17.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). *Dense*ly Connected Convolutional Networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 4700–4708. https://github.com/liuzhuang13/*Dense*Net.

Istiqamah, A. M. (2020). *Klasifikasi Citra Menggunakan Convolutional Neural Network Dengan Arsitektur Inception V4 Berbasis Android Pada Dataset Flower Recognition*.

Li, H., Wang, P., You, M., & Shen, C. (2018). Reading car license plates using deep neural networks. *Image and Vision Computing*, *72*, 14–23. https://doi.org/10.1016/j.imavis.2018.02.002

Luqman Hakim, Sari, Z., & Handhajani, H. (2021). Klasifikasi Citra Pigmen Kanker Kulit Menggunakan *Convolutional Neural Network*. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(2), 379–385. https://doi.org/10.29207/resti.v5i2.3001

Manasa, K., & Murthy, D. G. V. (2021). Skin Cancer Detection Using VGG-16. *European Journal of Molecular & Clinical Medicine*, *8*(1), 1419–1426.

Nuraeni, fitri, Yoga Handoko, A., & Nirwani Yusup, E. (2016). Aplikasi pakar untuk diagnosa penyakit kulit menggunakan metode forward chaining di al arif skin care kabupaten ciamis. *Seminar Nasional Teknologi Informasi Dan Multimedia*, 55–60.

Pardede, J., & Putra, D. A. L. (2020). Implementasi *Dense*Net Untuk Mengidentifikasi Kanker Kulit *Melanoma*. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(3). https://doi.org/10.28932/jutisi.v6i3.2814

Royana, F., Yuniar Maulida, P., Nurul Hasanah, R., & Setia Rahayu, S. (2021). Aplikasi *Mobile* Deteksi Dini Kanker Kulit Berdasarkan Image Processing. *Jurnal Litbang Edusaintech*, *2*(2), 100–106. http://journal.pwmjateng.com/index.php/jle

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv Preprint ArXiv*, 1409–1556. http://arxiv.org/abs/1409.1556

Suartika E.P, I. W., Wijaya, A. Y., & Soelaima, R. (2016). Klasifikasi Citra Menggunakan *Convolutional Neural Network* (Cnn) pada Caltech 101. *JURNAL TEKNIK ITS*, *5*, A65–A69.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going Deeper with Convolutions. *Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9. http://arxiv.org/abs/1409.4842

Wardhani, S. R. (2010). Biopsi dalam Bidang Dermatologi. *Maranatha Journal of Medicine and Health*, *5*(1), 14–23.

Wilvestra, S., Lestari, S., & Asri, E. (2018). Studi Retrospektif Kanker Kulit di Poliklinik Ilmu Kesehatan Kulit dan Kelamin RS Dr. M. Djamil Padang Periode Tahun 2015-2017. In *Jurnal Kesehatan Andalas* (Vol. 7). http://jurnal.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. v. (2017). *Learning Transferable Architectures for Scalable Image Recognition*. http://arxiv.org/abs/1707.07012

LAMPIRAN

1. Lampiran Code program

Berikut merupakan *Souce Code* yang digunakan pada pengimplementasian arsitektur CNN dalam mengklasifikasikan kanker kulit.

ln [1]: *import library*

```
import Numpy as np
import matplotlib.pyplot as plt
import cv2
import os
from os import listdir

import keras
from keras.preprocessing import image
from keras import backend as K
from keras.layers import Input
from keras.optimizers import Adam

from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from Sklearn.preprocessing import MultiLabelBinarizer
from Sklearn.preprocessing import LabelBinarizer
from Sklearn.model_selection import train_test_split

import tensorflow as tf
from tensorflow.keras import optimizers
from tensorflow.keras.models import Sequential, Model
from keras.layers import Conv2D, MaxPooling2D, Dense, Input,
Activation, Dropout, GlobalAveragePooling2D, \
    BatchNormalization, concatenate, AveragePooling2D, Flatten
```

Ln [2]: directory dataset

```
%cd "/content/drive/MyDrive/SkinCancer_10/cancer/actinic keratosis"
# read an image
img = cv2.imread('ISIC_0053826.jpg')
# show image format
print(img)
```

ln [3]: inisialisasi *hyperparameter*

```
EPOCHS = 100
INIT_LR = 1e-4
default_image_size = tuple((224, 224))
directory_root = "/content/drive/MyDrive/SKIN_CANCER"
width = 224
height = 224
depth = 3
```

ln [4]: Konversi Citra Menjadi Array

```
def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None :
            image = cv2.resize(image, default_image_size)
            return img_to_array(image)
        else :
            return np.array([])
    except Exception as e:
        print("Error :", e)
        return None
```

```
image_list, label_list = [], []
try:
    print("[INFO] Loading images ...")
    root_dir = listdir(directory_root)
    for directory in root_dir :
        # remove .DS_Store from list
        if directory == ".DS_Store" :
            root_dir.remove(directory)

    for Skin_folder in root_dir :

        Skin_cancer_folder_list =
listdir(f"{directory_root}/{Skin_folder}")

        for cancer_folder in Skin_cancer_folder_list :
            # remove .DS_Store from list
            if cancer_folder == ".DS_Store" :
                Skin_cancer_folder_list.remove(cancer_folder)

        for Skin_cancer_folder in Skin_cancer_folder_list:
            print(f"[INFO] Processing {Skin_cancer_folder} ...")
            Skin_cancer_folder_list =
listdir(f"{directory_root}/{Skin_folder}/{Skin_cancer_folder}/")

            for single_plant_disease_image in
Skin_cancer_folder_list :
                if single_plant_disease_image == ".DS_Store" :

Skin_cancer_folder_list.remove(single_plant_disease_image)

            for image in Skin_cancer_folder_list[:300]:
                image_directory =
f"{directory_root}/{Skin_folder}/{Skin_cancer_folder}/{image}"
                if image_directory.endswith(".jpg") == True or
image_directory.endswith(".JPG") == True:

image_list.append(convert_image_to_array(image_directory))
                    label_list.append(Skin_cancer_folder)
    print("[INFO] Image loading completed")
except Exception as e:
    print(f"Error : {e}")
```

ln [6]: pelabelan citra

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
n_classes = len(label_binarizer.classes_)
labels = os.listdir(directory_root)
print("Consist of " + str(n_classes) + " Classes")
```

ln [7]: normalisasi data citra

```
print("Normalizing data ..")
np_image_list = np.array(image_list, dtype=np.float16) / 225.0
```

ln [8]: *Splitting* data

```
x_train, x_test, y_train, y_test =
train_test_split(np_image_list,image_labels, test_size=0.2)
len(x_test), len(x_train)
```

ln [9]: augmetasi citra

```
aug = ImageDataGenerator(
    rotation_range=25,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    zoom_range=0.2,
    vertical_flip=True,
    horizontal_flip=True,
    fill_mode="nearest")
```

ln [10]: arsitektur model

```
#Arsitektur VGG16
import tensorflow as tf
from tensorflow.keras.applications.VGG16 import VGG16
model = tf.keras.Sequential([
    VGG16(
        include_top=False,
        weights='imagenet',
        input_shape=(224, 224, 3)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
#Arsitektur DenseNet121
import tensorflow as tf
from tensorflow.keras.applications.densenet import DenseNet121
model = tf.keras.Sequential([
    DenseNet121(
        include_top=False,
        weights='imagenet',
        input_shape=(224, 224, 3)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dropout(0.5),
```

```
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
#Arsitektur NASNetMobile
import tensorflow as tf
from tensorflow.keras.applications.nasnet import NASNetMobile
model = tf.keras.Sequential([
    NASNetMobile(
        include_top=False,
        weights='imagenet',
        input_shape=(224, 224, 3)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
optimizer = Adam(lr=0.0001)
model.compile(loss='categorical_crossentropy', optimizer=optimizer,
metrics=['accuracy'])
model.summary()
```

ln [12]: *training* model

```
%%time
history = model.fit(
    aug.flow(x_train, y_train, batch_size=64),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train) // 64,
    epochs=100,
    verbose=1)
```

```
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

ln [14]: menampilkan plot *accuracy*

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
#Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```

ln [15]: menampilkan *Confusion Matrix*

```
import seaborn as sns

ax = plt.subplot()
sns.heatmap(model_densenet, annot= True, ax = ax)

ax.set_xlabel('Predict labels')
ax.set_title('Confusion Matrix')
ax.xaxis.set_ticklabels(label_binarizer.classes_, rotation=90)
ax.yaxis.set_ticklabels(label_binarizer.classes_, rotation=0)
```

ln [16]: menampilkan kurva ROC

```
plt.figure(1, figsize=(10, 10))
plt.plot([0, 1], [0, 1], 'k--')
for i in range(len(label_binarizer.classes_)):
   fpr, tpr, thresholds = roc_curve(y_test[:, i], model_pred[:, i])
   individual_auc = auc(fpr, tpr)
   plt.plot(fpr, tpr, label= (label_binarizer.classes_[i] + '(area =
{})'.format(individual_auc)))

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC curve')
plt.legend(loc='best')
plt.show()
```

ln [17]: menampilakan *precision, Recall, F1-score*

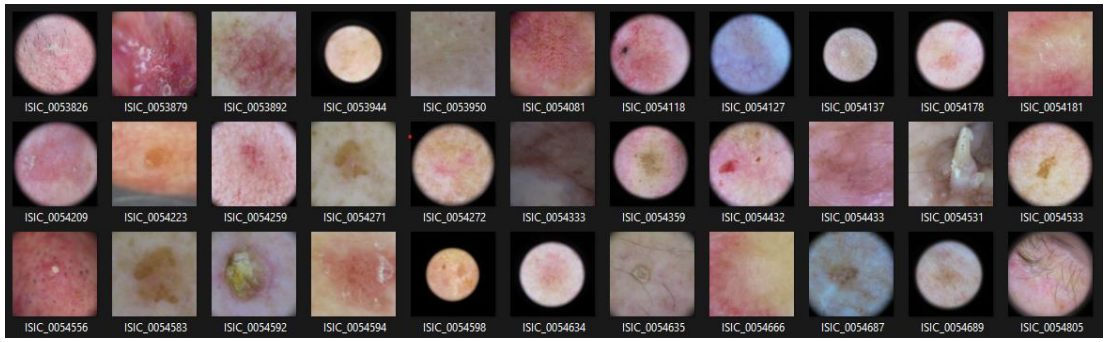
```
from Sklearn.metrics import classification_report
print (classification_report(y_test,y_pred))
```

source code Android

```
https://github.com/Ajrana/AppSkinCancer
```

2. Lampiran Gambar

Actinic keratosis

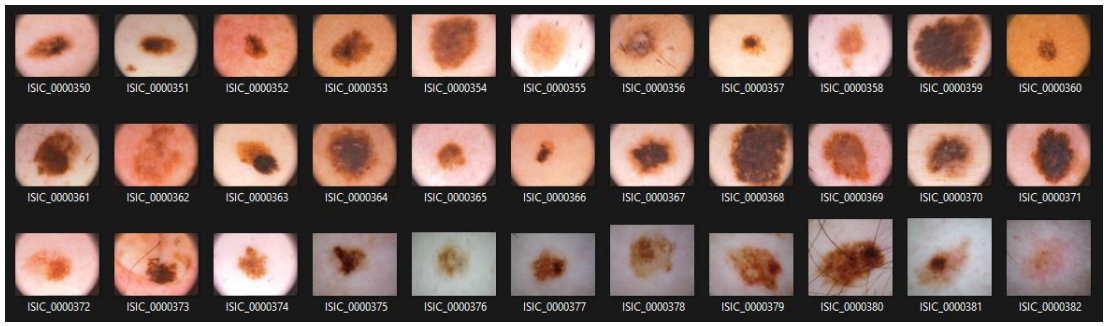


basal cell carcinoma

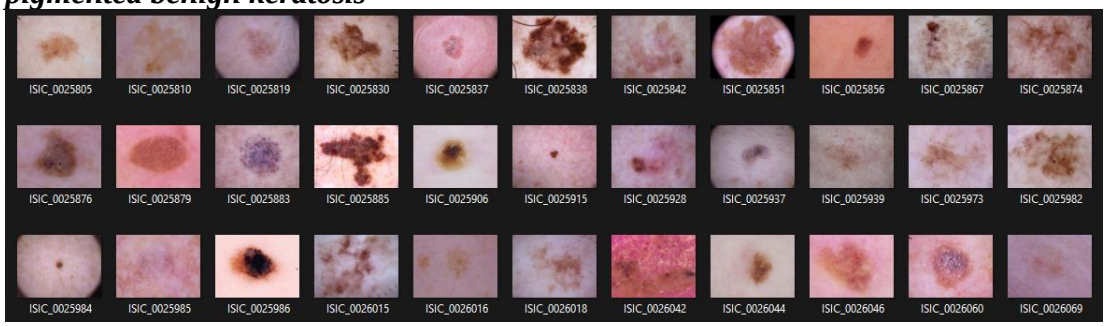
Dermatofibroma

melanoma

Nevus

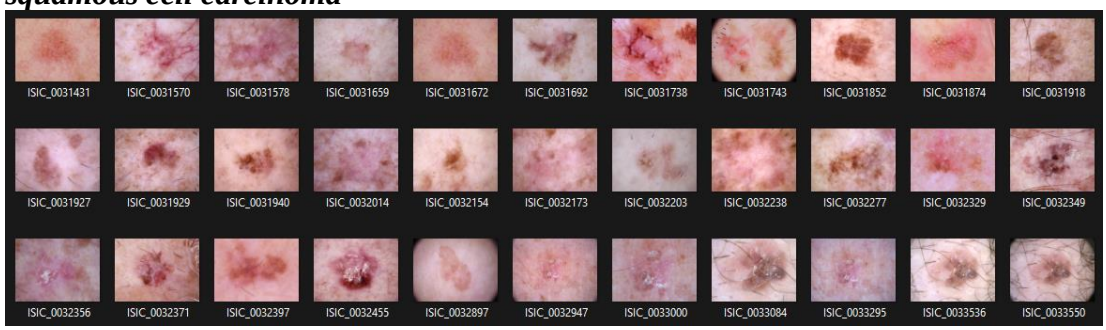


pigmented benign keratosis



seborrheic keratosis

squamous cell carcinoma



vascular lesion