DAFTAR PUSTAKA

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer International Publishing AG.

Arfan. (2018). Pallu Mara. Budaya Indonesia. Diakses pada 3 Juni 2023 dari https://budaya-indonesia.org/PALLU-MARA

Choudhary, K , DeCost, B., Chen, C. *et al.* (2022). *Recent advances and applications of deep learning methods in materials science*. npj Computational Materials. https://doi.org/10.1038/s41524-022-00734-6.

Darojat, M. D., Sari, Y. A., & Wihandika, R. C. (2021). *Convolutional Neural Network untuk Klasifikasi Citra Makanan Khas Indonesia*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 5(11), 4764-4769.

Fatmawaty, Nuddin, A., & Halimah, A. S. (2019). *Strategi Pengembangan Kualitas Kuliner Itik Palekko Di Rumah Makan Khas Bugis Di Kabupaten Sidrap*. Jurnal Pendidikan Teknologi Pertanian,S262-S270.

Feng, J., & Lu, S. (2019). *Performance Analysis of Various Activation Functions in Artificial Neural Networks*. Journal of Physics: Conference Series. doi:10.1088/1742-6596/1237/2/022030.

Geraldy, C., & Lubis, C. (2020). *Pendeteksian Dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once Dan Convolutional Neural Network*. Jurnal Ilmu Komputer dan Sistem Informasi, 8(2), 197-199.

Hafid, R. (2020). Barongko: Unggul pada Cita Rasa dan Nilai Filosofi. Indonesiana Platform Kebudayaan. Diakses pada 3 Juni 2023 dari https://kebudayaan.kemdikbud.go.id/bpnbsulsel/barongko-unggul-pada-cita-rasa-dan-nilai-filosofi/

Hariz, F. A., Yulita, I. N., & Suryana, I. (2022). *Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet*. Jurnal Ilmiah Teknologi Sistem Informasi, 3(4), 104-115.

Harsana, M., Triwidayati, M. (2020). *Potensi Makanan Tradisional Sebagai Daya Tarik Wisata Kuliner Di D.I. Yogyakarta*. Prosiding Pendidikan Teknik Boga Busana FT UNY, 15(1).

Indonesia Kaya. (2022). Es Pisang Ijo, Sesuap Sejarah Es Pisang Ijo. Diakses pada 3 Juni 2023 dari https://indonesiakaya.com/pustaka-indonesia/es-pisang-ijo/

Islam, M. T., Siddique, B. M. N. K., Rahman, S., & Jabid, T. (2018). *Food Image Classification with Convolutional Neural Network*. 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 257-262. doi: 10.1109/ICIIBMS.2018.8550005.

Juliansyah, S., & Laksito, A. D. (2021). *Klasifikasi Citra Buah Pir Menggunakan Convolutional Neural Networks*. InComTech: Jurnal Telekomunikasi dan Komputer, 11(1), 65-72.

Kholik, A. (2021). *Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram*. Jurnal Data Mining dan Sistem Informasi, 2(2), 10-20.

Lestari, N. K., Rahmanita, M., & Ingkadjaya, R. (2023). *Coto Mangkasara Sebagai Makanan Tradisional Dan Identitas Daerah Kota Makassar, Sulawesi Selatan*. Jurnal Industri Pariwisata, 5(2), 163-169.

Liu, H., & Lang, B. (2019). *Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey*. Applied Sciences, 9(20), 4396. doi:10.3390/app9204396.

Magdalena, R., Saidah, S., Pratiwi, N. K. C., & Putra, A. T. (2021). *Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN)*. Jurnal Edukasi dan Penelitian Informatika, 7(3), 335-339.

Noak, D. R. (2020). *Klasifikasi Citra Rontgen Gigi Berdasarkan Analisis Tekstur Untuk Diagnosis Karies Gigi Manusia Menggunakan Metode SVM*. Jurnal Elektronik Ilmu Komputer Udayana, 9(1), 141-150.

Musniar. (2008). Faktor Budaya, Sosial, Pribadi Dan Psikologis Dalam Hubungannya Dengan Keputusan Pembelian Kapurung (Tesis magister, Universitas Hasanuddin).

Nurchintiyawati, I. (2018). Pallu Basa. Budaya Indonesia. Diakses pada 3 Juni 2023 dari https://budaya-indonesia.org/PALLU-BASA/

Nurchintyawati, I. (2018). Mie Titi. Budaya Indonesia. Diakses pada 3 Juni 2023 dari https://budaya-indonesia.org/MIE-TITI/

Pallu Butung. (2022, 4 Maret). Dalam Wikipedia: Ensiklopedia Bebas. Diakses pada 3 Juni 2023 dari https://id.wikipedia.org/wiki/Pallu_butung

Pangestu, R. A., Rahmat, B., & Anggraeny, F. E. (2020). *Implementasi Algoritma Cnn Untuk Klasifikasi Citra Lahan Dan Perhitungan Luas*. Jurnal Informatika dan Sistem Informasi, 1(1), 166-174.

Peryanto, A., Yudhana, A., & Umar, R. (2019). *Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network*. Jurnal Ilmiah Teknik Informatika, 8(2), 138-147. http://dx.doi.org/10.22441/format.2019.v8.i2.007.

Portal Informasi Indonesia. (2019). Sop Konro Kuliner Legendaris Makassar. Diakses pada 3 Juni 2023 dari https://indonesia.go.id/kategori/pariwisata/945/sop-konro-kuliner-legendaris-makassar?lang=1

Prasmastio, R. M., Rahmat, B., & Yuniar, I. (2020). *Deteksi Dan Pengenalan Ikan Menggunakan Algoritma Convolutional Neural Network*. Jurnal Informatika dan Sistem Informasi, 1(2).

Qudsi, N. K., Asmara, R. A., & Syulistyo, A. R. (2019). *Identifikasi Citra Tulisan Tangan Digital Menggunakan Convolutional Neural Network (CNN)*. Prosiding Seminar Informatika Aplikatif Polinema.

Rabiah, S., Noer, M. U., & Ramlawati. (2016). *IBM Kampung Burasa' Sebagai Basis Kelompok Usaha Burasa' Di Kecamatan Pattallassang Kabupaten Gowa*. Jurnal Balireso, 1(1), 17-36.

Ramdani, A., Virgono, A., & Setianingsih, C. (2020). *Food Detecion With Image Processing Using Convolutional Neural Network (CNN) Method*. IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 91-96. doi:10.1109/IAICT50021.2020.9172024.

Rohayani, H., & Umam, M. C. (2022). *Prediksi Penentuan Program Studi Berdasarkan Nilai Siswa dengan Algoritma Backpropagation*. Journal of Information System Research, 3(4), 651-657. Doi: 10.47065/josh.v3i4.1935.

Saputra, I., & Kristiyanti, D. A. (2022). *Machine Learning Untuk Pemula*. Informatika.

Septhyan, S., Magdalena, R., & Pratiwi, N. K. C. (2022). *Deep Learning For The Detection Of Covid19, Pneumonia, And Tuberculosis In Chest X-Ray Imaging Using Cnn With Alexnet Architecture*. E-Proceeding of Engineering, 8(6), 2869-2878.

Suharni, Susilowati, E., Hidayat, T. M. (2021). *Implementasi Model Convolutional Neural Network (Cnn) Untuk Klasifikasi Penyakit Tbc Berbasis Dekstop*. UG Jurnal, 16(4).

Suta, I. B. L. M., Hartati, R. S., & Divayana, Y. (2019). *Diagnosa Tumor Otak Berdasarkan Citra MRI (Magnetic Resonance Imaging*. Majalah Ilmiah Teknologi Elektro, 18(2), 149-154. https://doi.org/10.24843/MITE.2019.v18i02.P01.

Sutoyo, E., & Fadlurrahman, M. A. (2020). *Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network*. Jurnal Edukasi dan Penelitian Informatika, 6(3), 379-385.

Toemon, S. (2017). Bobo Grid. Diakses pada 3 Juni 2023 dari https://bobo.grid.id/read/08675174/pisang-epe-makanan-khas-pantai-losari

Wahyuni, S., Sulaeman, M. (2022). *Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di PT Karya Komponen Presisi*. Jurnal Informatika SIMANTIK, 7(1), 12-21.

Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). *The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition.* Applied Sciences, 10(5), 1897. doi:10.3390/app10051897.

Wulandari, I., Yasin, H., & Widiharih, T. (2020). *Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (CNN)*. Jurnal Gaussian, 9(3), 273-282.

Yafis, B., & Rijal, H. (2023). *Analisa Respon Frekuensi Citra Digital Menggunakan Metode Transformasi Fourier Diskrit*. Jurnal Elektronika dan Teknologi Informasi, 4(1), 25-33.

Yuan , Z., Chen, G., Chen, Q., Li, W. & Long, S. (2020). *Breast Cancer Image Classification on CNN Classifier*. 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), 921-925. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00143

LAMPIRAN

```
# Load data
import os
main_path='d:/tika/Makanan Khas Sulsel/'
food=os.listdir(main_path)
print(food)

import cv2
import numpy as np
from tqdm import tqdm
IMG_SIZE=256
X_img=[]
Z_label=[]
def assign_label(img,food_type):
    return food_type

def make_train_data(food_type,DIR):
    for img in tqdm(os.listdir(DIR)):
        label=assign_label(img,food_type)
        path = os.path.join(DIR,img)
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))

        X_img.append(np.array(img))
        Z_label.append(str(label))

# Labelling & normalisasi piksel
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical
le=LabelEncoder()
Y=le.fit_transform(Z_label)
Y=to_categorical(Y,len(food))
X_img=np.array(X_img)
X_img=X_img/255

#pembagian data latih, data uji dan data validasi
from sklearn.model_selection import train_test_split
x_latih,x_uji,y_latih,y_uji=train_test_split(X_img,Y,test_size=0.10,random_state=7, stratify=Y)
x_latih,x_val,y_latih,y_val=train_test_split(x_latih,y_latih,test_size=0.15,random_state=7, stratify=y_latih)

# plot distribusi data
import pandas as pd
import matplotlib.pyplot as plt
import random as rn
train_counts = np.sum(y_latih, axis=0)
val_counts = np.sum(y_val, axis=0)
test_counts = np.sum(y_uji, axis=0)
food_chart=pd.DataFrame({'train': train_counts,'val': val_counts,'test': test_counts}, index=food)
x = np.arange(len(food))  # the label locations
width = 0.25  # the width of the bars
multiplier = 0
plt.figure(figsize=(15, 5))
for attribute, measurement in food_chart.items():
    offset = width * multiplier
    rects = plt.bar(x + offset, measurement, width, label=attribute)
    plt.bar_label(rects, padding=3)
    multiplier += 1
plt.xticks(x + width, food)
plt.xlabel('Kelas')
plt.ylabel('Jumlah Sampel')
```

```
plt.title('Distribusi Kelas pada Data Validasi')
plt.legend(loc='upper right')
plt.ylim(0, 85)
plt.show()


# model arsitektur CNN
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Activation, Dense
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, AveragePooling2D, Input
from keras.optimizers import Adam,SGD

models = Sequential()
models.add(Input(shape = (IMG_SIZE,IMG_SIZE,3)))
models.add(Conv2D(filters = 8, kernel_size = (3,3),padding = 'Same',activation ='relu'))
models.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

models.add(Conv2D(filters =16, kernel_size = (3,3),padding = 'Same',activation ='relu'))
models.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

models.add(Conv2D(filters =32, kernel_size = (3,3),padding = 'Same',activation ='relu'))
models.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

models.add(Conv2D(filters =64, kernel_size = (3,3),padding = 'Same',activation ='relu'))
models.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

models.add(Flatten())
models.add(Dense(128, activation ='relu'))
models.add(Dropout(0.25))
models.add(Dense(len(food), activation = "softmax"))

# augmentasi data
from tensorflow.keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(
        rotation_range=25,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.25, # Randomly zoom image
        width_shift_range=0.25,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.25,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True)  # randomly flip images


datagen.fit(x_latih)

# konfigurasi parameter
models.compile(optimizer=Adam(learning_rate=0.001),loss='categorical_crossentropy',metrics=['accuracy'])

# ringkasan model
models.summary()

# trainingb data
batch_size=32
epochs=200
History = models.fit(datagen.flow(x_latih,y_latih, batch_size=batch_size),
                             epochs = epochs, validation_data  = (x_val,y_val),
                             verbose = 1, steps_per_epoch=x_latih.shape[0] // batch_size)

# evaluasi validasi
model_eval=models.evaluate(x_val, y_val, return_dict=True)

# plot grafik akurasi dan loss
plt.plot(History.history['loss'])
plt.plot(History.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'val'])
plt.show()

plt.plot(History.history['accuracy'])
plt.plot(History.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'val'])
plt.show()

# Look at confusion matrix
import itertools
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score,precision_score,recall_score,confusion_matrix,roc_curve,roc_auc_score
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
```

```
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

# Predict the values from the testing dataset
Y_pred_ = models.predict(x_uji)
# Convert predictions classes to one hot vectors
Y_pred_classes_ = np.argmax(Y_pred_,axis = 1)
# Convert testing observations to one hot vectors
Y_true_ = np.argmax(y_uji,axis = 1)
# compute the confusion matrix
confusion_mtx_ = confusion_matrix(Y_true_, Y_pred_classes_)
# plot the confusion matrix
plot_confusion_matrix(confusion_mtx_, classes = (food))
```

```
# Metrik evaluasi
correct_ = np.nonzero(Y_pred_classes_==Y_true_)[0]
incorrect_ = np.nonzero(Y_pred_classes_!=Y_true_)[0]
from sklearn.metrics import classification_report
target_names = ["Class {}".format(i) for i in range(len(food))]
print(classification_report(Y_true_, Y_pred_classes_, target_names=target_names))

from sklearn.preprocessing import label_binarize
n_classes = len(food)
y_true_bin = label_binarize(Y_true_, classes=range(n_classes))

# Hitung ROC dan AUC untuk setiap kelas
fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_true_bin[:, i], Y_pred_[:, i])
    roc_auc[i] = roc_auc_score(y_true_bin[:, i], Y_pred_[:, i])

# Plot ROC curve untuk setiap kelas
plt.figure(figsize=(12, 8))
colors = plt.cm.jet(np.linspace(0, 1, n_classes))
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'ROC curve of class {i} (area = {roc_auc[i]:.2f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) for Multiclass')
plt.legend(loc='lower right', bbox_to_anchor=(1.05, 0))
plt.grid(True)
plt.show()
```