LAPORAN PROYEK SEGMENTASI U-NET UNTUK KANKER PAYUDARA

Nama : Zulfikar Juniarto

Npm : 22010022

# **PENDAHULUAN**

1. Latar Belakang

Kanker payudara adalah salah satu jenis kanker yang paling umum di dunia. Deteksi dini menggunakan USG (Ultrasonografi) sangat penting untuk meningkatkan tingkat kesembuhan pasien. Dalam proyek ini, digunakan model deep learning berbasis U-Net untuk melakukan segmentasi gambar kanker payudara guna membantu dalam diagnosis medis.

1. Tujuan
   * Mengembangkan model segmentasi berbasis U-Net untuk mendeteksi kanker payudara dari gambar USG.
   * Menggunakan dataset **Breast Ultrasound Images Dataset** dari Kaggle.
   * Melakukan evaluasi performa model menggunakan **akurasi, IoU (Intersection over Union), dan Dice Coefficient**.

# **METODOLOGI**

1. Dataset yang Digunakan

Dataset yang digunakan adalah Breast Ultrasound Images Dataset, yang terdiri dari:

* **Jinak (Benign):** Gambar USG dengan tumor jinak.
* **Ganas (Malignant):** Gambar USG dengan tumor ganas.
* **Normal:** Gambar USG tanpa indikasi kanker.  
  Setiap gambar memiliki pasangan **masker ground truth** yang menunjukkan area tumor

1. Dataset terdiri dari folder utama dengan subfolder berdasarkan kategori tumor:

Dataset\_BUSI\_with\_GT/

│── Benign/

│ ├── 1.png

│ ├── 1\_mask.png

│── Malignant/

│ ├── 2.png

│ ├── 2\_mask.png

│── Normal/

│ ├── 3.png

Setiap Gambar Memiki Mask yang menandai area Tumor.

# IMPLEMENTASI

1. Instalasi Library

pip install tensorflow numpy matplotlib opencv-python pandas kagglehub keras

1. Import Library

pip install tensorflow numpy matplotlib oimport tensorflow as tf

import numpy as np

import cv2

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from keras.models import Model

from keras.layers import Input, Conv2D, MaxPooling2D, Conv2DTranspose, concatenate

from keras.optimizers import Adampencv-python pandas kagglehub keras

1. Preprocessing Dataset

import kagglehub

breast\_ultrasound\_images\_dataset\_path = kagglehub.dataset\_download('aryashah2k/breast-ultrasound-images-dataset')

Dataset kemudia dipindahkan kedalam director kerja.

1. Fungsi Load Dataset

def load\_image(path, size):

image = cv2.imread(path)

image = cv2.resize(image, (size, size))

image = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)

image = image / 255.0 # Normalisasi

return image

mv /root/.cache/kagglehub/datasets/aryashah2k/breast-ultrasound-images-dataset/versions/1 /content/

Dataset kemudian dibagi menjadi **training (80%) dan testing (20%)**.

# PEMODELAN

* 1. Arsitektur U-Net

Model U-Net digunakan karena kemampuannya dalam segmentation tasks. U-Net terdiri dari dua bagian utama:

* **Encoder:** Menyusun fitur dengan convolutional layers.
* **Decoder:** Menggunakan **upsampling** untuk merekonstruksi gambar segmentasi.

Berikut adalah implementasi arsitektur U-Net:

def unet\_model(input\_size=(128, 128, 1)):

inputs = Input(input\_size)

# Encoder

conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)

conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv1)

pool1 = MaxPooling2D(pool\_size=(2, 2))(conv1)

conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool1)

conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv2)

pool2 = MaxPooling2D(pool\_size=(2, 2))(conv2)

# Decoder

up1 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(conv2)

merge1 = concatenate([conv1, up1], axis=3)

conv3 = Conv2D(64, (3, 3), activation='relu', padding='same')(merge1)

output = Conv2D(1, (1, 1), activation='sigmoid')(conv3)

model = Model(inputs=inputs, outputs=output)

model.compile(optimizer=Adam(lr=1e-4), loss='binary\_crossentropy', metrics=['accuracy'])

return model

* 1. Pelatihan Model

model = unet\_model()

history = model.fit(X\_train, y\_train, epochs=25, batch\_size=16, validation\_data=(X\_test, y\_test))

# EVALUASI MODEL

1. Evaluasi dengan Accuracy dan Loss

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('Model Accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.legend(['Train', 'Validation'])

plt.show()

1. Untuk mengevaluasi performa segmentasi, digunakan IoU dan Dice Coefficient:

def iou(y\_true, y\_pred):

intersection = np.sum(y\_true \* y\_pred)

union = np.sum(y\_true) + np.sum(y\_pred) - intersection

return intersection / union

def dice\_coef(y\_true, y\_pred):

intersection = np.sum(y\_true \* y\_pred)

return (2. \* intersection) / (np.sum(y\_true) + np.sum(y\_pred))

y\_pred = model.predict(X\_test)

iou\_score = iou(y\_test, y\_pred)

dice\_score = dice\_coef(y\_test, y\_pred)

print(f"IoU Score: {iou\_score:.4f}, Dice Coefficient: {dice\_score:.4f}")

# HASIL SEGMENTASI

* 1. Visualisasi Prediksi

plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)

plt.imshow(X\_test[i], cmap='gray')

plt.title('Original')

plt.subplot(1, 3, 2)

plt.imshow(y\_test[i], cmap='gray')

plt.title('Ground Truth')

plt.subplot(1, 3, 3)

plt.imshow(y\_pred[i], cmap='gray')

plt.title('Prediction')

plt.show()plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)

plt.imshow(X\_test[i], cmap='gray')

plt.title('Original')

plt.subplot(1, 3, 2)

plt.imshow(y\_test[i], cmap='gray')

plt.title('Ground Truth')

plt.subplot(1, 3, 3)

plt.imshow(y\_pred[i], cmap='gray')

plt.title('Prediction')

plt.show()

# KESIMPULAN

* **U-Net** berhasil digunakan untuk segmentasi kanker payudara dengan performa yang baik.
* IoU dan Dice Coefficient menunjukkan model dapat membedakan tumor dengan akurasi tinggi.
* Model dapat diimprovisasi lebih lanjut dengan **augmentation dan hyperparameter tuning**.