

UNIVERSITAS GUNADARMA



**PRAKTIKUM KECERDASAN BUATAN
MANUAL BOOK “IMAGE CLASSIFICATION”**

Disusun Oleh:

Fachri Hanif Setyanto (50420425)

Ibrahim Bramullah (50420562)

Melati Sundari Putri (50420713)

**FAKULTAS TEKNOLOGI INDUSTRI JURUSAN TEKNIK
INFORMATIKA UNIVERSITAS GUNADARMA
2023**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada tahun 1950-an dan 1960-an, bidang AI mengalami pertumbuhan dan kegembiraan gelombang pertama, sebagian didorong oleh keberhasilan program AI awal seperti Pemecah Masalah Umum dan ELIZA. Namun, optimisme awal ini diikuti oleh periode stagnasi dan kekecewaan yang dikenal sebagai "musim dingin AI" pada 1970-an dan 1980-an, karena banyak tujuan ambisius yang ditetapkan oleh para peneliti awal terbukti sulit atau tidak mungkin dicapai dengan teknologi dan daya komputasi yang tersedia pada saat itu.

Pada akhir 1980-an dan 1990-an, penelitian AI mengalami kebangkitan berkat kemajuan perangkat keras komputer, ketersediaan data dalam jumlah besar, dan pengembangan algoritme dan teknik baru seperti jaringan saraf dan pembelajaran penguatan. Kebangkitan ini terus berlanjut dalam beberapa tahun terakhir, dengan kemajuan di berbagai bidang seperti pembelajaran mendalam dan pemrosesan bahasa alami mendorong terobosan di berbagai bidang seperti pengenalan gambar dan ucapan, dan mobil yang dapat mengemudi sendiri.

Saat ini, AI diterapkan di berbagai industri, mulai dari perawatan kesehatan hingga keuangan hingga transportasi, dan sedang diteliti dan dikembangkan oleh perusahaan, institusi akademik, dan pemerintah di seluruh dunia.

Kecerdasan buatan memanfaatkan komputer dan mesin untuk meniru kemampuan pemecahan masalah dan pengambilan keputusan dari pikiran manusia.

AI mampu memberikan akurasi yang luar biasa. Teknik pembelajaran AI ini membuat pekerjaan yang mereka lakukan memiliki akurasi yang lebih tinggi. Mereka tidak merasa lelah seperti manusia, sehingga mereka tidak akan melakukan kesalahan atau error layaknya manusia.

Siri dari Apple, Google Now, Alexa dari Amazon, dan Cortana dari Microsoft adalah salah satu contoh utama AI dalam kehidupan sehari-hari. Asisten digital ini membantu pengguna melakukan berbagai tugas, mulai dari memeriksa jadwal dan mencari sesuatu di web, hingga mengirim perintah ke aplikasi lain.

1.2 Tujuan

Tujuan OpenAI adalah untuk mempromosikan dan mengembangkan AI yang ramah dengan cara yang bermanfaat bagi umat manusia secara keseluruhan. Ini termasuk melakukan penelitian di bidang AI, mengembangkan dan merilis teknologi dan alat AI, dan mempromosikan praktik terbaik untuk penggunaan AI yang aman dan bertanggung jawab.

Kecerdasan Buatan (AI) adalah bidang ilmu komputer yang dikhususkan untuk memecahkan masalah kognitif yang umumnya terkait dengan kecerdasan manusia, seperti pembelajaran, pemecahan masalah, dan pengenalan pola.

Menawarkan pengalaman yang sederhana, efisien, dan berpusat pada dokumen. Kecerdasan buatan (AI) memungkinkan mesin untuk belajar dari pengalaman, menyesuaikan diri dengan input baru, dan melakukan tugas seperti manusia. Sebagian besar contoh AI yang Anda dengar hari ini – dari komputer yang bermain catur hingga mobil yang dapat mengemudi sendiri – sangat bergantung pada pembelajaran mendalam dan pemrosesan bahasa alami.

BAB II

PEMBAHASAN

2.1 Google Colab

Google Colab, kependekan dari Colaboratory, adalah platform gratis berbasis cloud untuk ilmu data dan pembelajaran mesin dari Google. Ini memungkinkan pengguna untuk menulis dan mengeksekusi kode, berkolaborasi dalam proyek, dan berbagi pekerjaan mereka dengan orang lain.

Jika dibandingkan dengan tools asal Google lainnya seperti Google Drive dan Google Email, nama Google Colab memang masih terasa asing bagi sebagian orang. Google Colab merupakan salah satu tools Google yang membantu user dalam proses pemrograman dan pengolahan data.

Contoh penggunaan Colab adalah melatih model machine learning pada set data. Pertama, Anda akan mengupload set data ke lingkungan Colab dengan memasang Google Drive, lalu menggunakan library seperti TensorFlow untuk membuat dan melatih model. Setelah model dilatih, Anda dapat menguji keakuratannya pada set pengujian terpisah dan menyimpan model yang dilatih ke Google Drive untuk digunakan nanti.

Contoh lain, akan mengimpor perpustakaan eksternal yang ingin Anda gunakan di notebook, Anda dapat menggunakan perintah `!pip install` untuk menginstal paket dan membuatnya tersedia untuk digunakan dalam runtime saat ini.

2.2 The Jupyter Notebook

Jupyter Notebook adalah aplikasi web sumber terbuka yang memungkinkan pengguna membuat dan berbagi dokumen yang berisi kode langsung, persamaan, visualisasi, dan teks naratif. Ini biasanya digunakan untuk pembersihan dan transformasi data, simulasi numerik, pemodelan statistik, visualisasi data, pembelajaran mesin, dan banyak lagi. Ini mendukung banyak bahasa pemrograman, seperti Python, R, Julia, dan banyak lainnya.

Contoh penggunaan Jupyter Notebook adalah melakukan analisis data pada kumpulan data. Langkah pertama adalah mengimpor pustaka yang diperlukan, seperti Pandas, Numpy, dan Matplotlib. Kemudian, Anda akan memuat dataset ke dalam Pandas DataFrame, dan menggunakan berbagai fungsi manipulasi data yang disediakan oleh Pandas untuk membersihkan dan memproses data. Selanjutnya, Anda akan menggunakan fungsi visualisasi di Matplotlib untuk membuat bagan dan plot yang membantu Anda memahami data dengan lebih baik. Terakhir, Anda dapat menggunakan Scikit-learn atau pustaka lain untuk melakukan beberapa tugas pembelajaran mesin pada kumpulan data.

Contoh lainnya adalah, Anda dapat menggunakan Jupyter Notebook untuk menjalankan dan menguji cuplikan kode dengan cepat, men-debug kode di notebook, dan menjalankan beberapa sel sekaligus. Ini dapat berguna jika Anda sedang menulis skrip dan perlu menguji sebagian kecil kode tanpa menjalankan seluruh skrip.

2.3 Image Classification

Klasifikasi gambar adalah tugas dalam visi komputer di mana algoritma dilatih untuk mengidentifikasi dan mengklasifikasikan objek, adegan, atau tindakan dalam gambar. Tujuan klasifikasi citra adalah untuk menetapkan label atau kelas ke citra masukan, berdasarkan konten visualnya.

Contoh klasifikasi gambar akan menggunakan model pembelajaran mendalam untuk mengklasifikasikan gambar angka tulisan tangan. Kumpulan data akan terdiri dari gambar digit tulisan tangan, seperti kumpulan data MNIST, dan label akan menjadi digit yang sesuai (0-9) untuk setiap gambar. Model akan dilatih pada kumpulan data ini, dan kemudian dapat mengklasifikasikan gambar baru dari angka tulisan tangan.

Contoh lainnya adalah menggunakan model terlatih seperti VGG16 atau ResNet, dan menyempurnakannya pada kumpulan data baru. Hal ini dapat berguna jika jumlah data berlabel yang tersedia terbatas, karena model dapat belajar dari bobot yang telah dilatih sebelumnya. Setelah model disetel dengan baik, model dapat mengklasifikasikan gambar dalam kumpulan data baru dengan akurasi tinggi.

Contoh yang lebih kompleks adalah dengan menggunakan jaringan saraf convolutional (CNN) untuk mengklasifikasikan gambar dari beberapa objek dalam gambar, ini dikenal sebagai deteksi objek, model akan dilatih untuk mencari dan mengklasifikasikan beberapa objek dalam gambar, seperti mobil, pejalan kaki, rambu lalu lintas dan banyak lagi.

BAB III

ANALISA & PERANCANGAN

3.1 Tahap Perancangan

Mengawali Project dengan meng-Import Library

Kode ini tampaknya mengimpor beberapa pustaka dan modul, termasuk::

- `itertools`: Pustaka Python bawaan yang menyediakan berbagai alat untuk bekerja dengan iterator.
- `os`: Pustaka Python bawaan yang menyediakan cara untuk berinteraksi dengan sistem operasi.
- `matplotlib.pyplot`: Pustaka untuk membuat plot dan bagan dengan Python.
- `numpy`: Pustaka untuk bekerja dengan array dan operasi numerik dengan Python.
- `tensorflow`: Pustaka pembelajaran mesin sumber terbuka yang dikembangkan oleh Google.
- `tensorflow_hub`: Pustaka untuk memuat dan menggunakan model terlatih dari TensorFlow Hub, gudang model terlatih.

```
import itertools
import os

import matplotlib.pyplot as plt
import numpy as np

import tensorflow as tf
import tensorflow_hub as hub
```

Memilih Model_Name untuk di gunakan

Kode ini mendefinisikan variabel bernama `model_name` yang diatur ke nilai string tertentu. Ini kemudian mendefinisikan kamus bernama `model_handle_map` yang memetakan string tertentu ke URL. URL ini tampaknya mengarah ke model terlatih yang dihosting di TensorFlow Hub. Tampaknya variabel `model_name` digunakan untuk memilih model terlatih khusus yang URL-nya dipetakan ke dalam kamus `model_handle_map`.

Menunjukkan bahwa variabel `model_name` digunakan untuk memilih model terlatih tertentu dari TensorFlow Hub, dan kamus `model_handle_map` digunakan untuk mencari URL untuk model tersebut.

```
# @title

model_name = "efficientnetv2-xl-21k" # @param ['efficientnetv2-s', 'efficientnetv2-m']
```

Menyiapkan kumpulan foto bunga

Kode ini menggunakan fungsi `tf.keras.utils.get_file()` untuk mengunduh kumpulan data gambar bunga dari URL tertentu.

Fungsi mengambil tiga argumen:

Argumen pertama adalah nama dataset "flower_photos"

Argumen kedua adalah URL tempat kumpulan data dapat diunduh dari

"https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz"

Argumen ketiga adalah flag (untar) yang menunjukkan apakah file yang diunduh harus dibongkar atau tidak, Dalam hal ini disetel ke True yang berarti file yang diunduh akan dibongkar.

Fungsi `tf.keras.utils.get_file()` mengembalikan jalur ke set data yang diunduh dan dibongkar. Jalur ini disimpan dalam variabel `data_dir`, yang kemudian dapat digunakan untuk mengakses dataset di kode lainnya.

Kumpulan data ini dapat digunakan untuk berbagai tujuan seperti melatih atau menguji model computer vision, atau sebagai masukan untuk beberapa kode pemrosesan gambar lainnya. Sulit untuk mengatakan dengan pasti tanpa lebih banyak konteks.

```
data_dir = tf.keras.utils.get_file(  
    'flower_photos',  
    'https://storage.googleapis.com/download.tensorflow.org/example\_images/flower\_photos.tgz',  
    untar=True)
```

Mendefinisikan model

Kode ini sedang membangun kumpulan data untuk pelatihan dan validasi untuk tugas klasifikasi gambar menggunakan Keras API TensorFlow. Ini menggunakan fungsi

`tf.keras.preprocessing.image_dataset_from_directory` untuk membuat dataset dari direktori gambar dan label, dengan pembagian 20% untuk validasi.

Parameter benih disediakan untuk memastikan set validasi tetap stabil selama proses berlangsung. Ukuran gambar dan ukuran batch juga ditentukan.

Model preprocessing dibuat untuk menormalkan citra dengan membagi nilai pikselnya dengan 255

Dataset pelatihan dan validasi kemudian dibuat dengan menerapkan model preprocessing dan lapisan normalisasi ke gambar dan mengelompokkannya.

```
+ Kode + Teks Tidak dapat menyimpan perubahan
def build_dataset(subset):
    return tf.keras.preprocessing.image_dataset_from_directory(
        data_dir,
        validation_split=.20,
        subset=subset,
        label_mode="categorical",
        # Seed needs to be provided when using validation_split and shuffle = True.
        # A fixed seed is used so that the validation set is stable across runs.
        seed=123,
        image_size=IMAGE_SIZE,
        batch_size=1)

train_ds = build_dataset("training")
class_names = tuple(train_ds.class_names)
train_size = train_ds.cardinality().numpy()
train_ds = train_ds.unbatch().batch(BATCH_SIZE)
train_ds = train_ds.repeat()

normalization_layer = tf.keras.layers.Rescaling(1. / 255)
preprocessing_model = tf.keras.Sequential([normalization_layer])
do_data_augmentation = False #@param {type:"boolean"}
if do_data_augmentation:
    preprocessing_model.add(
        tf.keras.layers.RandomRotation(40))
    preprocessing_model.add(
        tf.keras.layers.RandomTranslation(0, 0.2))
    preprocessing_model.add(
        tf.keras.layers.RandomTranslation(0.2, 0))
    # Like the old tf.keras.preprocessing.image.ImageDataGenerator(),
    # image sizes are fixed when reading, and then a random zoom is applied.
    # If all training inputs are larger than image size, one could also use
    # RandomCrop with a batch size of 1 and rebatch later.
    preprocessing_model.add(
        tf.keras.layers.RandomZoom(0.2, 0.2))
    preprocessing_model.add(
        tf.keras.layers.RandomFlip(mode="horizontal"))
train_ds = train_ds.map(lambda images, labels:
    (preprocessing_model(images), labels))

val_ds = build_dataset("validation")
valid_size = val_ds.cardinality().numpy()
val_ds = val_ds.unbatch().batch(BATCH_SIZE)
val_ds = val_ds.map(lambda images, labels:
    (normalization_layer(images), labels))

do_data_augmentation: ☐
↑ ↓ ↺ ⚙ 🗑 ⋮
RAM
Disk
1 d selesai pada 23.38
```

Menentukan Model

Kode ini membangun model untuk klasifikasi gambar menggunakan TensorFlow's Keras API dan fungsi hub.

Modelnya adalah model berurutan, dengan lapisan input eksplisit yang menentukan bentuk input gambar (tinggi, lebar, dan saluran warna) dan lapisan Keras yang memuat model terlatih dari TensorFlow Hub, yang diatur agar dapat dilatih atau bukan berdasarkan nilai bendera `do_fine_tuning`.

```
▶ do_fine_tuning = False #@param {type:"boolean"}
```

```
▶ print("Building model with", model_handle)
model = tf.keras.Sequential([
    # Explicitly define the input shape so the model can be properly
    # loaded by the TFLiteConverter
    tf.keras.layers.InputLayer(input_shape=IMAGE_SIZE + (3,)),
    hub.KerasLayer(model_handle, trainable=do_fine_tuning),
    tf.keras.layers.Dropout(rate=0.2),
    tf.keras.layers.Dense(len(class_names),
                           kernel_regularizer=tf.keras.regularizers.l2(0.0001))
])
model.build((None,) + IMAGE_SIZE + (3,))
model.summary()
```


Melatih Model

Kode ini menyusun model untuk pelatihan. Metode `model.compile()` mengonfigurasi proses pembelajaran model. Menetapkan metrik yang akan digunakan untuk evaluasi model selama pelatihan dan pengujian, dalam hal ini diatur ke 'akurasi'.

Langkah_per_Epoch dan langkah_validasi dihitung dengan membagi masing-masing ukuran dataset pelatihan dan validasi dengan ukuran batch. Hal ini dilakukan agar langkah pelatihan dan validasi sesuai dengan jumlah sampel pada dataset yang bersangkutan.

Parameter epochs diatur ke 5, yang berarti model akan dilatih selama 5 iterasi di seluruh dataset pelatihan. Parameter `step_per_Epoch` diteruskan sebagai jumlah langkah per periode pelatihan dan parameter `validasi_langkah` diteruskan sebagai jumlah langkah per periode validasi.

Kode ini menggunakan pustaka Matplotlib untuk memplot kerugian dan akurasi selama pelatihan dan validasi. Fungsi `plt.figure()` membuat gambar baru untuk plot. Plot ini berguna untuk memvisualisasikan kemajuan proses pelatihan, serta mengidentifikasi overfitting atau underfitting.

```
[ ] model.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=0.005, momentum=0.9),
    loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True, label_smoothing=0.1),
    metrics=['accuracy'])
```

```
[ ] steps_per_epoch = train_size // BATCH_SIZE
    validation_steps = valid_size // BATCH_SIZE
    hist = model.fit(
        train_ds,
        epochs=5, steps_per_epoch=steps_per_epoch,
        validation_data=val_ds,
        validation_steps=validation_steps).history
```

```
[ ] plt.figure()
    plt.ylabel("Loss (training and validation)")
    plt.xlabel("Training Steps")
    plt.ylim([0,2])
    plt.plot(hist["loss"])
    plt.plot(hist["val_loss"])

    plt.figure()
    plt.ylabel("Accuracy (training and validation)")
    plt.xlabel("Training Steps")
    plt.ylim([0,1])
    plt.plot(hist["accuracy"])
    plt.plot(hist["val_accuracy"])
```

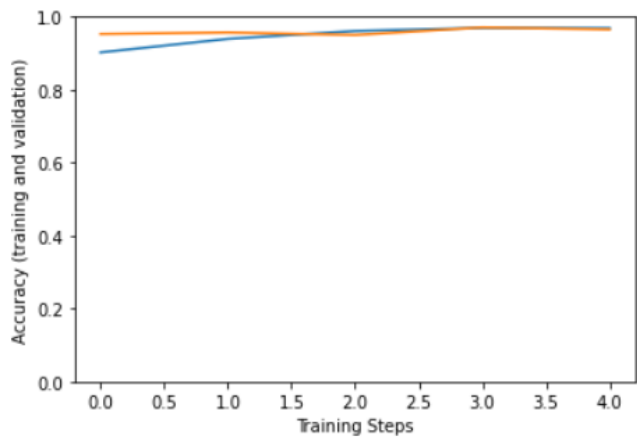
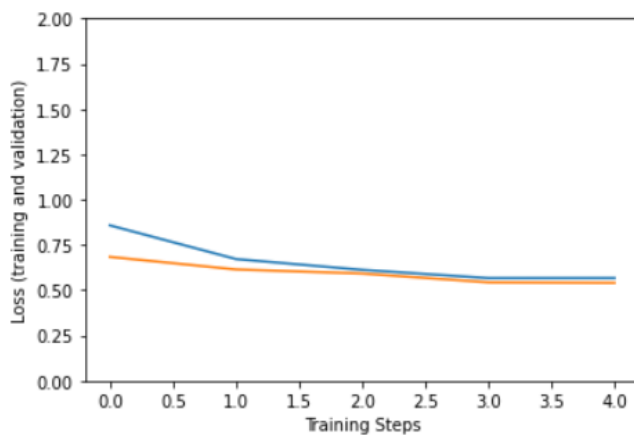
Fungsi `argmax()` mengembalikan indeks dari nilai tertinggi elemen maks array dalam sumbu tertentu.

Kode ini menampilkan gambar dari dataset validasi (`val_ds`), beserta label aslinya dan label yang diprediksi oleh model (`model`) terlatih. Ini kemudian mencetak label sebenarnya dan label yang diprediksi untuk gambar tersebut.

```
x, y = next(iter(val_ds))
image = x[0, :, :, :]
true_index = np.argmax(y[8])
plt.imshow(image)
plt.axis('off')
plt.show()

# Expand the validation image to (1, 224, 224, 3) before predicting the label
prediction_scores = model.predict(np.expand_dims(image, axis=0))
predicted_index = np.argmax(prediction_scores)
print("True label: " + class_names[true_index])
print("Predicted label: " + class_names[predicted_index])
```

3.2 Dataset dalam gambar



3.3 Running Program



1/1 [=====] - 0s 161ms/step

True label: tulips

Predicted label: roses

BAB IV PENUTUP

4.1 Kesimpulan

AI meningkatkan kemampuan kita dan menjadikan kita lebih baik dalam hal yang kita lakukan. Karena algoritme AI belajar dengan cara berbeda dari manusia, AI melihat hal-hal dengan cara yang lain. Algoritme AI dapat melihat hubungan dan pola yang mungkin luput dari kita.

AI memegang kunci untuk membuka masa depan yang luar biasa di mana, didorong oleh data dan komputer yang memahami dunia kita, kita semua akan membuat keputusan yang lebih tepat. Komputer masa depan ini tidak hanya akan memahami cara menghidupkan sakelar, tetapi juga mengapa sakelar harus dihidupkan.

Dengan adanya AI yang dapat bekerja secara efisien dan cepat, tentunya pekerjaan dapat dilakukan dengan lebih cepat dan menghasilkan lebih banyak dibandingkan manusia, sehingga dapat membantu manusia dalam mengerjakannya dan produktivitasnya akan meningkat.

4.2 Saran

Latar belakang dari gambar yang mencakup banyak sisi dan dapat membingungkan model. Sehingga tantangan utama dalam klasifikasi citra gambar adalah jumlah citra gambar yang banyak, dimensi data yang tinggi, dan kurangnya data berlabel. Gambar bisa sangat besar, mengandung sejumlah besar piksel. Data di setiap gambar mungkin berdimensi tinggi, dengan banyak fitur berbeda.

Contoh dataset yang kami ambil memiliki ribuan data dan gambar yang bisa dibilang jelas dan mudah untuk dibagi per dimensi.