Лабораторна робота 8 ІАД

Завантаження набору даних

Почнемо із завантаження та розпакування архіву з даними:

```
import tensorflow as tf
import zipfile
import os

# Завантаження даних
dataset_url = "https://storage.googleapis.com/ztm_tf_course/food_vision/101_food_classes_10_percent.zip"
dataset_path = tf.keras.utils.get_file("food101.zip", dataset_url, extract=False)

# Розпакування
extraction_dir = "food101_data"
with zipfile.ZipFile(dataset_path, 'r') as zip_ref:
    zip_ref.extractall(extraction_dir)

print(f"Дані розпаковано в папку: {extraction_dir}")
```

Вибір класів для класифікації

Визначимо три класи відповідно до формули:

```
# Визначення індексів класів
n = 10  # Замініть на ваш номер у списку
i1 = n - 1
i2 = n + 29
i3 = n + 59

# Отримання назв класів
data_dir = os.path.join(extraction_dir, "101_food_classes_10_percent/train")
all_classes = sorted(os.listdir(data_dir))

# Вибір класів
selected_classes = [all_classes[i] for i in [i1, i2, i3]]
print(f"Вибрані класи: {selected_classes}")
```

Завантаження даних

Ми завантажимо лише вибрані класи з набору даних:

```python
from tensorflow.keras.preprocessing.image_dataset import image_dataset_from_directory

# Завантаження тренувального та тестового набору даних
train_dataset = image_dataset_from_directory(
    data_dir,
    labels="inferred",
    label_mode="int",
    class_names=selected_classes,
    batch_size=32,
    image_size=(224, 224),
    shuffle=True
)

val_dataset = image_dataset_from_directory(
    os.path.join(extraction_dir, "101_food_classes_10_percent/test"),
    labels="inferred",
    label_mode="int",
    class_names=selected_classes,
    batch_size=32,
    image_size=(224, 224),
    shuffle=True
)

print("Дані завантажено.")
```

Аугментація даних

Додамо випадкові трансформації для покращення якості моделі:

```python
data_augmentation = tf.keras.Sequential([
    tf.keras.layers.RandomFlip("horizontal"),
    tf.keras.layers.RandomRotation(0.2),
    tf.keras.layers.RandomZoom(0.2)
])
```

Побудова моделі

Використаємо **MobileNetV2** як базову модель:

```python
base_model = tf.keras.applications.MobileNetV2(include_top=False)
base_model.trainable = False  # Заморожуємо базову модель

# Створення повної моделі
model = tf.keras.Sequential([
    data_augmentation,
    tf.keras.layers.Rescaling(1./255),  # Нормалізація пікселів
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(3, activation='softmax')  # 3 класи
])

# Компіляція моделі
model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)

model.summary()
```

Оцінка результатів

1. **Графіки навчання**:

```python
import matplotlib.pyplot as plt

# Графік точності
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title("Графік точності")
plt.legend()
plt.show()

# Графік втрат
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title("Графік втрат")
plt.legend()
plt.show()
```

Оцінка точності на тестових даних:

```python
loss, accuracy = model.evaluate(val_dataset)
print(f"Точність моделі: {accuracy * 100:.2f}%")
```

Прогнозування для нових зображень

Завантажте зображення та виконайте прогноз:

```python
from tensorflow.keras.utils import load_img, img_to_array
import numpy as np

# Завантаження зображення
img_path = "path_to_your_image.jpg"  # Змініть шлях до вашого зображення
img = load_img(img_path, target_size=(224, 224))
img_array = img_to_array(img) / 255.0
img_array = tf.expand_dims(img_array, 0)  # Додаємо вимір для батча

# Прогноз
predictions = model.predict(img_array)
predicted_class = selected_classes[np.argmax(predictions)]
print(f"Передбачений клас: {predicted_class}")
```

Висновок

- **Точність моделі:** Результат буде залежати від обраних класів та кількості епох.
- **Модель:** Навчена нейронна мережа для класифікації трьох класів.
- **Прогнозування:** Можна перевірити якість класифікації на нових зображеннях.