In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, Input, Reshape
from tensorflow.keras.utils import to_categorical
```

In [2]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\sathwik\anaconda3\lib\site-packages (2.17.0)
Requirement already satisfied: tensorflow-intel==2.17.0 in c:\users\sathwik\anaconda3\lib\site-packages (from ten
sorflow) (2.17.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-in
tel==2.17.0->tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow
-intel==2.17.0->tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorf
low-intel==2.17.0->tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\sathwik\anaconda3\lib\site-package
s (from tensorflow-intel==2.17.0->tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorfl
ow-intel==2.17.0->tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-inte
l==2.17.0->tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-
intel==2.17.0->tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in c:\users\sathwik\anaconda3\lib\site-packages (from tens
orflow-intel==2.17.0->tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow
-intel==2.17.0->tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-intel==
2.17.0->tensorflow) (23.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 i
n c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (4.25.5)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorfl
ow-intel==2.17.0->tensorflow) (2.31.0)
Requirement already satisfied: setuptools in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-intel=
=2.17.0->tensorflow) (68.0.0)
Requirement already satisfied: six>=1.12.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-intel
==2.17.0->tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-
intel==2.17.0->tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\sathwik\anaconda3\lib\site-packages (from ten
sorflow-intel==2.17.0->tensorflow) (4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-int
el==2.17.0->tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorfl
ow-intel==2.17.0->tensorflow) (1.67.0)
Requirement already satisfied: tensorboard<2.18,>=2.17 in c:\users\sathwik\anaconda3\lib\site-packages (from tens
orflow-intel==2.17.0->tensorflow) (2.17.1)
Requirement already satisfied: keras>=3.2.0 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorflow-inte
l==2.17.0->tensorflow) (3.6.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\sathwik\anaconda3\lib\site-packag
es (from tensorflow-intel==2.17.0->tensorflow) (0.31.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorf
low-intel==2.17.0->tensorflow) (1.24.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\sathwik\anaconda3\lib\site-packages (from astunpars
e>=1.6.0->tensorflow-intel==2.17.0->tensorflow) (0.38.4)
Requirement already satisfied: rich in c:\users\sathwik\anaconda3\lib\site-packages (from keras>=3.2.0->tensorflo
w-intel==2.17.0->tensorflow) (13.9.2)
Requirement already satisfied: namex in c:\users\sathwik\anaconda3\lib\site-packages (from keras>=3.2.0->tensorfl
ow-intel==2.17.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in c:\users\sathwik\anaconda3\lib\site-packages (from keras>=3.2.0->tensorf
low-intel==2.17.0->tensorflow) (0.13.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sathwik\anaconda3\lib\site-packages (from req
uests<3,>=2.21.0->tensorflow-intel==2.17.0->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\sathwik\anaconda3\lib\site-packages (from requests<3,>=2.
21.0->tensorflow-intel==2.17.0->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sathwik\anaconda3\lib\site-packages (from requests<
3,>=2.21.0->tensorflow-intel==2.17.0->tensorflow) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sathwik\anaconda3\lib\site-packages (from requests<
3,>=2.21.0->tensorflow-intel==2.17.0->tensorflow) (2023.11.17)
Requirement already satisfied: markdown>=2.6.8 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorboard<
2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\sathwik\anaconda3\lib\site-packa
ges (from tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorboard<
2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\sathwik\anaconda3\lib\site-packages (from werkzeug>=
1.0.1->tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (2.1.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\sathwik\anaconda3\lib\site-packages (from rich->
keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\sathwik\anaconda3\lib\site-packages (from rich
->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\sathwik\anaconda3\lib\site-packages (from markdown-it-py>=
2.2.0->rich->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (0.1.0)
```

In [3]:
```
import tensorflow as tf
```

In [4]:
```
pip install tensorflow
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sathwik\anaconda3\lib\site-packages (from request
s<3,>=2.21.0->tensorflow-intel==2.17.0->tensorflow) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sathwik\anaconda3\lib\site-packages (from request
s<3,>=2.21.0->tensorflow-intel==2.17.0->tensorflow) (2023.11.17)
Requirement already satisfied: markdown>=2.6.8 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorboar
d<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\sathwik\anaconda3\lib\site-pac
kages (from tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\sathwik\anaconda3\lib\site-packages (from tensorboar
d<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\sathwik\anaconda3\lib\site-packages (from werkzeug
>=1.0.1->tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0->tensorflow) (2.1.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\sathwik\anaconda3\lib\site-packages (from rich
->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\sathwik\anaconda3\lib\site-packages (from ri
ch->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\sathwik\anaconda3\lib\site-packages (from markdown-it-py>
=2.2.0->rich->keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (0.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
# Paths to the folders
license_detection_folder = r"C:\Users\sathwik\soulpage\Licplatesdetection_train\license_plates_detection_train"
license_recognition_folder=r"C:\Users\sathwik\soulpage\Licplatesrecognition_train\license_plates_recognition_train
test_folder=r"C:\Users\sathwik\soulpage\test\test\test"
license_detection_csv=r"C:\Users\sathwik\soulpage\Licplatesdetection_train (1).csv"
license_recognition_csv=r"C:\Users\sathwik\soulpage\Licplatesrecognition_train.csv"
```

In [6]:
```
# Load the annotations
license_detection_df = pd.read_csv(license_detection_csv)
license_recognition_df = pd.read_csv(license_recognition_csv)
```

Data Preprocessing

In [13]:
```
# Load images and preprocess
def load_images_from_folder(folder, image_size=(224, 224)):
    images = []
    filenames = []
    for filename in os.listdir(folder):
        if filename.endswith('.jpg'):
            img = cv2.imread(os.path.join(folder, filename))
            if img is not None:
                img = cv2.resize(img, image_size)
                images.append(img)
                filenames.append(filename)
    return np.array(images), filenames

# license_detection_images, license_detection_filenames = C:\Users\sathwik\soulpage(license_detection_folder)
# license_recognition_images, license_recognition_filenames = load_images_from_folder(license_recognition_folder)
# test_images, test_filenames = load_images_from_folder(test_folder)
```

In [14]:
```
license_detection_images, license_detection_filenames = load_images_from_folder(license_detection_folder)
license_recognition_images, license_recognition_filenames = load_images_from_folder(license_recognition_folder)
test_images, test_filenames = load_images_from_folder(test_folder)
```

In [15]:
```
# Normalize images
license_detection_images = license_detection_images / 255.0
license_recognition_images = license_recognition_images / 255.0
test_images = test_images / 255.0
```

In [16]:
```
# Ensure filenames match between CSV and image folder
license_detection_df['img_id'] = license_detection_df['img_id'].apply(lambda x: x.split('/')[-1])
license_recognition_df['img_id'] = license_recognition_df['img_id'].apply(lambda x: x.split('/')[-1])

# Filter dataframes to only include rows with existing images
license_detection_df = license_detection_df[license_detection_df['img_id'].isin(license_detection_filenames)]
license_recognition_df = license_recognition_df[license_recognition_df['img_id'].isin(license_recognition_filename

# Ensure consistent length
assert len(license_detection_df) == len(license_detection_images)
assert len(license_recognition_df) == len(license_recognition_images)
```

In [17]:
```
# Split license detection data into training and validation sets
train_images, val_images, train_labels, val_labels = train_test_split(
    license_detection_images, license_detection_df, test_size=0.2, random_state=42)
```

Model Detection

In [18]:
```
# Load pre-trained model and add custom layers

base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(4, activation='linear')(x)  # 4 outputs for bounding box coordinates

model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weig
hts_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5 (https://storage.googleapis.com/tensorflow/keras-applications/mo
bilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5)
9406464/9406464 ━━━━━━━━━━━━━━━━━━━━ 8s 1us/step
```

In [19]:
```
# Convert DataFrame to numpy array for bounding box labels

train_labels = train_labels[['ymin', 'xmin', 'ymax', 'xmax']].values.astype(np.float32)
val_labels = val_labels[['ymin', 'xmin', 'ymax', 'xmax']].values.astype(np.float32)
```

In [20]:
```
# Train the model
history = model.fit(train_images, train_labels, validation_data=(val_images, val_labels), epochs=10, batch_size=32
```

```
Epoch 1/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 114s 4s/step - loss: 84260.2812 - val_loss: 219717.6875
Epoch 2/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 74s 3s/step - loss: 9340.9395 - val_loss: 255494.8594
Epoch 3/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 73s 3s/step - loss: 5094.7046 - val_loss: 246962.8906
Epoch 4/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 75s 3s/step - loss: 2730.9717 - val_loss: 227442.1719
Epoch 5/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 73s 3s/step - loss: 1660.4061 - val_loss: 212870.0469
Epoch 6/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 75s 3s/step - loss: 1344.6694 - val_loss: 242122.5781
Epoch 7/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 72s 3s/step - loss: 1078.4769 - val_loss: 301786.8438
Epoch 8/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 72s 3s/step - loss: 1214.0692 - val_loss: 213386.3594
Epoch 9/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 76s 3s/step - loss: 1005.0651 - val_loss: 211595.5625
Epoch 10/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 73s 3s/step - loss: 1088.6036 - val_loss: 211855.0156
```

4.Model Recognition

In [21]:
```
# Define the model
recognition_model = Sequential([
    Input(shape=(224, 224, 3)),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10 * 37, activation='softmax'),  # Flattened one-hot vectors, 37 for 0-9, A-Z and padding
    Reshape((10, 37))  # Assuming max 10 characters in license plate
])

# Compile the model
recognition_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Prepare labels for recognition model
def prepare_labels(df, image_filenames, max_length=10):
    labels = []
    for filename in image_filenames:
        text = df[df['img_id'] == filename]['text'].values[0]
        label = [ord(char) - ord('0') if char.isdigit() else ord(char) - ord('A') + 10 for char in text]
        # Pad label to max_length with 36 (for padding)
        label += [36] * (max_length - len(label))
        labels.append(label)
    return np.array(labels)

recognition_labels = prepare_labels(license_recognition_df, license_recognition_filenames)
```

In [22]:
```
# One-hot encode labels
num_classes = 36  # 10 digits + 26 letters
recognition_labels = np.array([to_categorical(label, num_classes + 1) for label in recognition_labels])
```

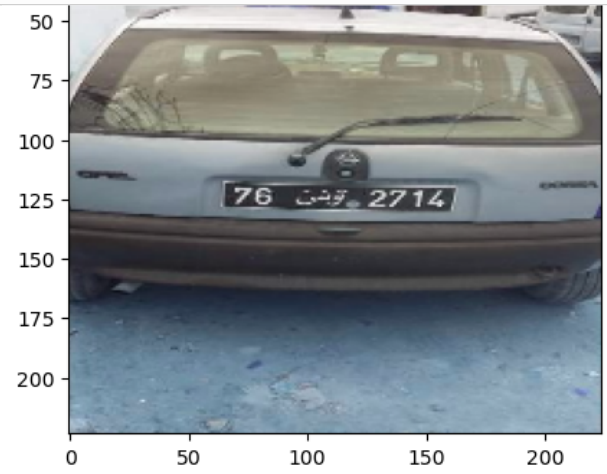
In [23]:
```
# Split data into training and validation sets
train_images, val_images, train_labels, val_labels = train_test_split(
    license_recognition_images, recognition_labels, test_size=0.2, random_state=42)
```

In [24]:
```
# Train the recognition model
recognition_history = recognition_model.fit(train_images, train_labels, validation_data=(val_images, val_labels),
```

```
Epoch 1/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 25s 960ms/step - accuracy: 0.2531 - loss: 3.0682 - val_accuracy: 0.4600 - val_loss: 1.
7963
Epoch 2/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 17s 759ms/step - accuracy: 0.4278 - loss: 2.0095 - val_accuracy: 0.4600 - val_loss: 1.
6472
Epoch 3/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 18s 767ms/step - accuracy: 0.4437 - loss: 1.8066 - val_accuracy: 0.4589 - val_loss: 1.
6325
Epoch 4/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 20s 890ms/step - accuracy: 0.4519 - loss: 1.7340 - val_accuracy: 0.4678 - val_loss: 1.
5867
Epoch 5/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 20s 830ms/step - accuracy: 0.4651 - loss: 1.6747 - val_accuracy: 0.4722 - val_loss: 1.
5566
Epoch 6/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 17s 750ms/step - accuracy: 0.4837 - loss: 1.5800 - val_accuracy: 0.4894 - val_loss: 1.
4896
Epoch 7/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 17s 739ms/step - accuracy: 0.5159 - loss: 1.4617 - val_accuracy: 0.5006 - val_loss: 1.
4776
Epoch 8/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 18s 778ms/step - accuracy: 0.5464 - loss: 1.3777 - val_accuracy: 0.5106 - val_loss: 1.
4263
Epoch 9/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 19s 809ms/step - accuracy: 0.5562 - loss: 1.2954 - val_accuracy: 0.5217 - val_loss: 1.
4079
Epoch 10/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 21s 904ms/step - accuracy: 0.5833 - loss: 1.2392 - val_accuracy: 0.5106 - val_loss: 1.
4180
```

5.Prediction and Accuracy

In [25]:
```
# Ensure val_labels and val_images have correct shapes
print(f"Validation images shape: {val_images.shape}")
print(f"Validation labels shape: {val_labels.shape}")

# Evaluate the recognition model
recognition_loss, recognition_accuracy = recognition_model.evaluate(val_images, val_labels)
print(f"Recognition Model Loss: {recognition_loss}")
print(f"Recognition Model Accuracy: {recognition_accuracy}")

# Make predictions
recognition_predictions = recognition_model.predict(test_images)

# Decode recognition predictions
def decode_predictions(preds, max_length=100):
    decoded_texts = []
    for pred in preds:
        decoded_text = ''.join([chr(np.argmax(char) + ord('0')) if np.argmax(char) < 100 else chr(np.argmax(char)
        decoded_texts.append(decoded_text.replace(chr(num_classes + ord('0')), ''))
    return decoded_texts

decoded_texts = decode_predictions(recognition_predictions)

# Display some test results
for i in range(5):
    plt.imshow(test_images[i])
    plt.title(f"Predicted: {decoded_texts[i]}")
    plt.show()
```



In [26]:
```
import math
num_samples = 10
rows = math.ceil(num_samples / 5)
plt.figure(figsize=(20, 4 * rows))
for i in range(num_samples):
    plt.subplot(rows, 5, i + 1)
    plt.imshow(test_images[i])
    plt.title(f"Predicted: {decoded_texts[i]}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```

In [27]:
```
# Save predictions to submission file
submission_df = pd.DataFrame({'filename': test_filenames, 'text': decoded_texts})
submission_df.to_csv('submission.csv', index=False)
```

In [30]:
```
submission_df.head(20)
```

Out[30]:

	filename	text
0	1000.jpg	13MM898
1	1001.jpg	13MM593
2	1002.jpg	137M293
3	1003.jpg	11MM215
4	1004.jpg	13MM690
5	1005.jpg	13MM908
6	1006.jpg	17MM908
7	1007.jpg	17MM598
8	1008.jpg	13MM992
9	1009.jpg	17MM593
10	1010.jpg	13MM293
11	1011.jpg	13MM293
12	1012.jpg	17MM998
13	1013.jpg	17MM898
14	1014.jpg	13MM493
15	1015.jpg	13MM508
16	1016.jpg	13MM993
17	1017.jpg	13MM798
18	1018.jpg	17MM598
19	1019.jpg	17MM498

In [ ]:
```

```