In [5]:
```
from warnings import filterwarnings
filterwarnings(action="ignore")

import os
import cv2
import copy
import numpy as np
import pandas as pd
from PIL import Image
from tqdm import tqdm
from glob import glob as glb
from xml.etree import ElementTree

import pytesseract

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

import torch
from torch import nn
from torch.utils.data import Dataset, DataLoader
from torch.optim.lr_scheduler import ReduceLROnPlateau

import albumentations as A

from torchvision.models import mobilenet_v2

import yaml
import shutil

try:
    from ultralytics import YOLO
except:
    !pip install ultralytics
    from ultralytics import YOLO

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

print("Successfully loaded libraries")
```

```
Collecting ultralytics
  Downloading ultralytics-8.2.28-py3-none-any.whl.metadata (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.2/41.2 kB 2.0 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=3.3.0 in /opt/conda/lib/python3.10/site-p
ackages (from ultralytics) (3.7.5)
Requirement already satisfied: opencv-python>=4.6.0 in /opt/conda/lib/python3.10/sit
e-packages (from ultralytics) (4.9.0.80)
Requirement already satisfied: pillow>=7.1.2 in /opt/conda/lib/python3.10/site-packa
ges (from ultralytics) (9.5.0)
Requirement already satisfied: pyyaml>=5.3.1 in /opt/conda/lib/python3.10/site-packa
ges (from ultralytics) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in /opt/conda/lib/python3.10/site-pa
ckages (from ultralytics) (2.31.0)
Requirement already satisfied: scipy>=1.4.1 in /opt/conda/lib/python3.10/site-packag
es (from ultralytics) (1.11.4)
Requirement already satisfied: torch>=1.8.0 in /opt/conda/lib/python3.10/site-packag
es (from ultralytics) (2.1.2)
Requirement already satisfied: torchvision>=0.9.0 in /opt/conda/lib/python3.10/site-
packages (from ultralytics) (0.16.2)
Requirement already satisfied: tqdm>=4.64.0 in /opt/conda/lib/python3.10/site-packag
es (from ultralytics) (4.66.4)
Requirement already satisfied: psutil in /opt/conda/lib/python3.10/site-packages (fr
om ultralytics) (5.9.3)
Requirement already satisfied: py-cpuinfo in /opt/conda/lib/python3.10/site-packages
(from ultralytics) (9.0.0)
Requirement already satisfied: pandas>=1.1.4 in /opt/conda/lib/python3.10/site-packa
ges (from ultralytics) (2.2.1)
Requirement already satisfied: seaborn>=0.11.0 in /opt/conda/lib/python3.10/site-pac
kages (from ultralytics) (0.12.2)
Collecting ultralytics-thop>=0.2.5 (from ultralytics)
  Downloading ultralytics_thop-0.2.7-py3-none-any.whl.metadata (8.5 kB)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.10/site-pa
ckages (from matplotlib>=3.3.0->ultralytics) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-packag
es (from matplotlib>=3.3.0->ultralytics) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.10/site-p
ackages (from matplotlib>=3.3.0->ultralytics) (4.47.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.10/site-p
ackages (from matplotlib>=3.3.0->ultralytics) (1.4.5)
Requirement already satisfied: numpy<2,>=1.20 in /opt/conda/lib/python3.10/site-pack
ages (from matplotlib>=3.3.0->ultralytics) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-pac
kages (from matplotlib>=3.3.0->ultralytics) (21.3)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.10/site-pa
ckages (from matplotlib>=3.3.0->ultralytics) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.10/sit
e-packages (from matplotlib>=3.3.0->ultralytics) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packag
es (from pandas>=1.1.4->ultralytics) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-pack
ages (from pandas>=1.1.4->ultralytics) (2023.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.1
0/site-packages (from requests>=2.23.0->ultralytics) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packag
es (from requests>=2.23.0->ultralytics) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-
```

```
packages (from requests>=2.23.0->ultralytics) (1.26.18)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-
packages (from requests>=2.23.0->ultralytics) (2024.2.2)
Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages
(from torch>=1.8.0->ultralytics) (3.13.1)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.10/site-p
ackages (from torch>=1.8.0->ultralytics) (4.9.0)
Requirement already satisfied: sympy in /opt/conda/lib/python3.10/site-packages (fro
m torch>=1.8.0->ultralytics) (1.12)
Requirement already satisfied: networkx in /opt/conda/lib/python3.10/site-packages
(from torch>=1.8.0->ultralytics) (3.2.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.10/site-packages (fr
om torch>=1.8.0->ultralytics) (3.1.2)
Requirement already satisfied: fsspec in /opt/conda/lib/python3.10/site-packages (fr
om torch>=1.8.0->ultralytics) (2024.3.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages
(from python-dateutil>=2.7->matplotlib>=3.3.0->ultralytics) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-pac
kages (from jinja2->torch>=1.8.0->ultralytics) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /opt/conda/lib/python3.10/site-packag
es (from sympy->torch>=1.8.0->ultralytics) (1.3.0)
Downloading ultralytics-8.2.28-py3-none-any.whl (779 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 779.6/779.6 kB 20.3 MB/s eta 0:00:0000:0
1
Downloading ultralytics_thop-0.2.7-py3-none-any.whl (25 kB)
WARNING: Error parsing requirements for aiohttp: [Errno 2] No such file or director
y: '/opt/conda/lib/python3.10/site-packages/aiohttp-3.9.1.dist-info/METADATA'
Installing collected packages: ultralytics-thop, ultralytics
Successfully installed ultralytics-8.2.28 ultralytics-thop-0.2.7
Successfully loaded libraries
```

In [6]:
```python
def set_random_seed(seed):
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

seed = 42
set_random_seed(seed)
```

In [25]:
```python
def LoadData(image_path):
    data = {"imagepath":[], "xmin":[], "xmax":[], "ymin":[], "ymax":[], "width":[],
    file_path = glb(image_path+"/*.xml")
    for xml_path in file_path:
        root_element = ElementTree.parse(xml_path).getroot()
        imagepath = os.path.join(image_path, root_element.find("filename").text)
        bounding_box = root_element.find("object").find("bndbox")
        size = root_element.find("size")
        data["imagepath"].append(imagepath)
        data["xmin"].append(int(bounding_box.find("xmin").text))
        data["xmax"].append(int(bounding_box.find("xmax").text))
        data["ymin"].append(int(bounding_box.find("ymin").text))
        data["ymax"].append(int(bounding_box.find("ymax").text))
        data["width"].append(int(size.find("width").text))
```

```
        data["height"].append(int(size.find("height").text))
    return pd.DataFrame(data)
```

In [39]:
```
file_pth = "/kaggle/input/number-plate-detection/images"

df = LoadData(file_pth)

print(df.shape)

df.head()
```

```
(225, 7)
```

Out[39]:

	imagepath	xmin	xmax	ymin	ymax	width	height
0	/kaggle/input/number-plate-detection/images/N1...	244	369	240	293	600	430
1	/kaggle/input/number-plate-detection/images/N1...	331	538	263	317	850	681
2	/kaggle/input/number-plate-detection/images/N1...	80	335	150	243	445	349
3	/kaggle/input/number-plate-detection/images/N2...	131	209	129	153	350	196
4	/kaggle/input/number-plate-detection/images/N1...	180	559	216	314	678	475

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
imagepath    0
xmin         0
xmax         0
ymin         0
ymax         0
width        0
height       0
dtype: int64
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 225 entries, 0 to 224
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   imagepath  225 non-null    object
 1   xmin       225 non-null    int64 
 2   xmax       225 non-null    int64 
 3   ymin       225 non-null    int64 
 4   ymax       225 non-null    int64 
 5   width      225 non-null    int64 
 6   height     225 non-null    int64 
dtypes: int64(6), object(1)
memory usage: 12.4+ KB
```

In [7]:
```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
xmin	225.0	294.151111	272.849733	1.0	134.0	192.0	346.0	1804.0
xmax	225.0	500.942222	364.583056	94.0	290.0	373.0	568.0	2493.0
ymin	225.0	303.213333	243.297160	2.0	167.0	230.0	332.0	1734.0
ymax	225.0	363.177778	261.941062	50.0	213.0	288.0	397.0	1882.0
width	225.0	798.315556	574.606700	249.0	500.0	600.0	806.0	3720.0
height	225.0	548.244444	377.284787	166.0	337.0	425.0	572.0	2472.0

In [8]:
```
def PlotImage(index):
    fig, ax = plt.subplots(1, 3, figsize=(10,10))
    image = np.array(Image.open(df.loc[index, "imagepath"]))
    xmin, xmax, ymin, ymax = df.loc[index, ["xmin", "xmax", "ymin", "ymax"]].values

    ax[0].imshow(image)
    ax[0].set_title("Original car image")
    ax[0].axis("off")

    cv2.rectangle(image,(xmin, ymin), (xmax, ymax),(0,255,0),3)

    ax[1].imshow(image)
    ax[1].set_title("Detected Licence plate")
    ax[1].axis("off")

    roi = image[ymin:ymax, xmin:xmax]

    ax[2].imshow(roi)
    ax[2].set_title("Extracted Licence plate")
    ax[2].axis("off")

    extracted_character = pytesseract.image_to_string(roi)
    print(f"Extracted character -> {extracted_character}")

    plt.show()
```

In [9]:
```
PlotImage(100)
```

```
Extracted character -> Dente
```

In [10]:
```
class ImageData(Dataset):
    def __init__(self, df, aug=None, norm_label:bool=False):
        self.df = df
        self.norm_label = norm_label
        self.aug = aug if aug else A.Compose([A.Resize(height=224, width=224), A.No

    def __len__(self):
        return len(self.df)

    def __getitem__(self, index):
        try:
            loaded_image = np.array(Image.open(self.df["imagepath"].iloc[index]).co
            h, w, c = loaded_image.shape
            augmented_image = self.aug(image=loaded_image)["image"]
            if not self.norm_label:
                normalized_coordinates = np.array([self.df["xmin"].iloc[index],
                                                   self.df["xmax"].iloc[index],
                                                   self.df["ymin"].iloc[index],
                                                   self.df["ymax"].iloc[index]])
            else:
                normalized_coordinates = np.array([self.df["xmin"].iloc[index]/w,
                                                     self.df["xmax"].iloc[index]/w,
                                                     self.df["ymin"].iloc[index]/h,
                                                     self.df["ymax"].iloc[index]/h])

        except:
            raise RuntimeError(f"Error loading image at index {index}")

        return torch.permute(torch.tensor(augmented_image, dtype=torch.float32), (2
```

In [11]:
```
xtrain, xval = train_test_split(df, test_size=.2, random_state=True)

print(xtrain.shape, xval.shape)

xtrain.reset_index(drop=True, inplace=True)
xval.reset_index(drop=True, inplace=True)
```

```
(180, 7) (45, 7)
```

In [12]:
```
scaler = StandardScaler()

scaled_train = pd.DataFrame(scaler.fit_transform(xtrain.loc[:, ["xmin", "xmax", "ym
                            columns=xtrain.loc[:, ["xmin", "xmax", "ymin", "ymax"]]

scaled_val = pd.DataFrame(scaler.fit_transform(xval.loc[:, ["xmin", "xmax", "ymin",
                          columns=xval.loc[:, ["xmin", "xmax", "ymin", "ymax"]].col

Xtrain = pd.concat([xtrain["imagepath"], scaled_train], axis=1)
Xval = pd.concat([xval["imagepath"], scaled_val], axis=1)
```

In [13]:
```
Xtrain.isnull().sum()
```

Out[13]:
```
imagepath    0
xmin         0
xmax         0
ymin         0
ymax         0
dtype: int64
```

In [14]:
```
Xval.isnull().sum()
```

Out[14]:
```
imagepath    0
xmin         0
xmax         0
ymin         0
ymax         0
dtype: int64
```

In [15]:
```
train_data = ImageData(df=Xtrain)
val_data = ImageData(df=Xval)

train_dataloader = DataLoader(train_data, batch_size=13, shuffle=True)
val_dataloader = DataLoader(val_data, batch_size=13, shuffle=True)

print("Data Loader successfully created")
```

```
Data Loader successfully created
```

Modelling

In [16]:
```
base_model = mobilenet_v2(pretrained=True)

for param in base_model.parameters():
    if isinstance(param, nn.BatchNorm2d):
        param.requires_grad = False

base_model.classifier = nn.Sequential(
                            nn.Dropout(p=0.2, inplace=False),
                            nn.Linear(in_features=1280, out_features=4, bias=True)
                            )

base_model
```

```
Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to /roo
t/.cache/torch/hub/checkpoints/mobilenet_v2-b0353104.pth
100%|████████| 13.6M/13.6M [00:01<00:00, 8.46MB/s]
```

Out[16]:
```
MobileNetV2(
  (features): Sequential(
    (0): Conv2dNormActivation(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=F
alse)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), g
roups=32, bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), g
roups=96, bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (3): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
```

```
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
```

```
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (12): InvertedResidual(
```

```
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (13): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (14): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
```

```
stats=True)
      )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (18): Conv2dNormActivation(
      (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      (2): ReLU6(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=4, bias=True)
  )
)
```

In [17]:
```python
class EarlyStopping:
    def __init__(self, patience=2, min_delta=0, restore_best_weights=True):
        self.patience = patience
        self.min_delta = min_delta
        self.restore_best_weights = restore_best_weights
        self.best_model_wts = None
        self.best_loss = float('inf')
        self.counter = 0

    def __call__(self, model, val_loss):
        if self.best_loss - val_loss > self.min_delta:
            self.best_loss = val_loss
            self.counter = 0
            self.best_model_wts = copy.deepcopy(model.state_dict())
        else:
            self.counter += 1
            if self.counter >= self.patience:
                return True
        return False
```

In [18]:
```python
class DataTrainer():
    def __init__(self, model, dataloader: list, optimizer, loss_function, lr_schedu
        self.model = model.to("cuda:0" if torch.cuda.is_available() else "cpu")
        self.dataloader = dataloader
        self.optim = optimizer
        self.loss_func = loss_function
        self.lr_scheduler = lr_scheduler
        self.early_stopping = early_stopping
        self.device = "cuda:0" if torch.cuda.is_available() else "cpu"
        self.best_model_wts = copy.deepcopy(self.model.state_dict())
        self.best_val_loss = float('inf')

    def train_model(self):
        training_loss = 0

        self.model.train()
```

```
        for data in tqdm(self.dataloader[0], total=len(self.dataloader[0])):
            X, y = data[0].to(self.device), data[1].to(self.device)
            self.optim.zero_grad()
            model_output = self.model(X)
            model_loss = self.loss_func(model_output, y)
            training_loss += model_loss.item()
            model_loss.backward()
            self.optim.step()

        final_train_loss = training_loss / len(self.dataloader[0])
        return final_train_loss

    def validate_model(self):
        val_loss = 0

        self.model.eval()

        with torch.no_grad():
            for data in tqdm(self.dataloader[1], total=len(self.dataloader[1])):
                X, y = data[0].to(self.device), data[1].to(self.device)
                output = self.model(X)
                model_loss = self.loss_func(output, y)
                val_loss += model_loss.item()

        final_val_loss = val_loss / len(self.dataloader[1])
        return final_val_loss

    def fit(self, epochs: int):
        train_loss_per_epoch = []
        val_loss_per_epoch = []

        for epoch in range(epochs):
            print(f"Epoch: {epoch + 1}/{epochs}")
            train_loss = self.train_model()
            val_loss = self.validate_model()

            print(f"Training Loss --> {train_loss:.4f} Val Loss --> {val_loss:.4f}"

            train_loss_per_epoch.append(train_loss)
            val_loss_per_epoch.append(val_loss)

            # Step the scheduler if provided
            if self.lr_scheduler:
                if isinstance(self.lr_scheduler, torch.optim.lr_scheduler.ReduceLRO
                    self.lr_scheduler.step(val_loss)
                else:
                    self.lr_scheduler.step()

            # Check for early stopping
            if self.early_stopping and self.early_stopping(self.model, val_loss):
                print(f"Early stopping at epoch {epoch + 1}")
                break

            # Save best model weights
            if val_loss < self.best_val_loss:
```

```
                self.best_val_loss = val_loss
                self.best_model_wts = copy.deepcopy(self.model.state_dict())

        # Load the best model weights if early stopping is used and restore_best_we
        if self.early_stopping and self.early_stopping.restore_best_weights:
            print("Returning Best Model Weights")
            self.model.load_state_dict(self.early_stopping.best_model_wts)
        else:
            self.model.load_state_dict(self.best_model_wts)

        history = pd.DataFrame({
            "train_loss": train_loss_per_epoch,
            "val_loss": val_loss_per_epoch
        })

        return {"model": self.model, "history": history}
```

In [19]:
```
class RMSELoss(nn.Module):
    def __init__(self):
        super(RMSELoss, self).__init__()

    def forward(self, y_pred, y_true):
        mse = nn.MSELoss()(y_pred, y_true)
        rmse = torch.sqrt(mse)
        return rmse
```

In [20]:
```
loss_func = RMSELoss()
optimizers = torch.optim.Adam(params=base_model.parameters(), lr=0.001)
scheduler = ReduceLROnPlateau(optimizers, mode='min', factor=0.1, patience=2, verbo
early_stopping = EarlyStopping(patience=10, min_delta=0.01, restore_best_weights=Tr

trainer = DataTrainer(base_model,
                      dataloader=[train_dataloader, val_dataloader],
                      optimizer=optimizers,
                      loss_function=loss_func,
                      lr_scheduler=scheduler,
                      early_stopping=early_stopping)
```

In [21]:
```
# Fit the model
model_result = trainer.fit(epochs=50)
```

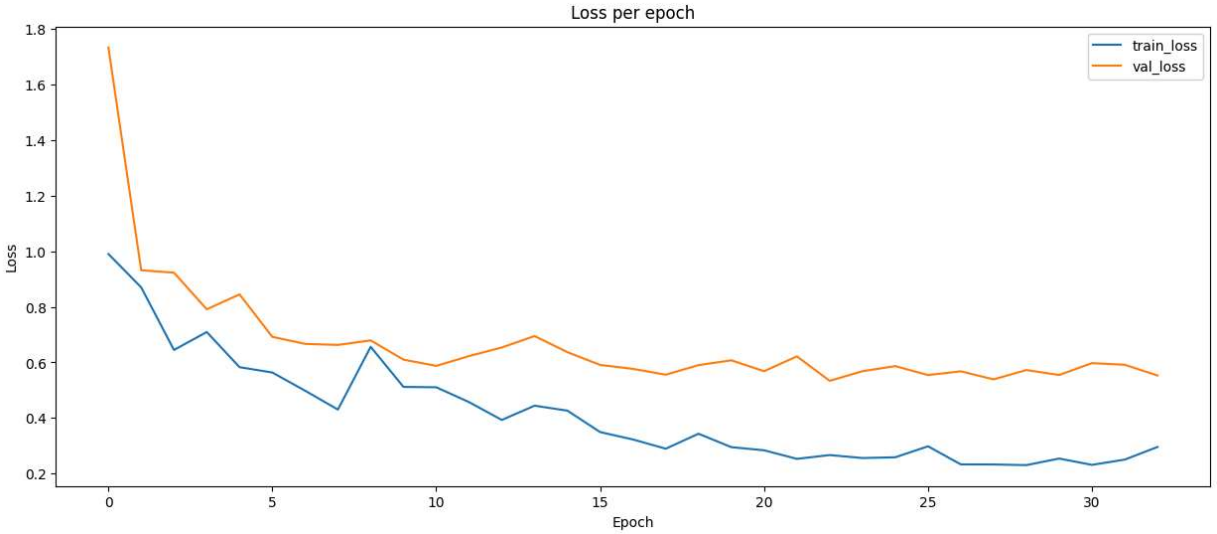
```
Epoch: 1/50
100%|██████████| 14/14 [00:04<00:00,  3.50it/s]
100%|██████████| 4/4 [00:00<00:00,  6.48it/s]
Training Loss --> 0.9902 Val Loss --> 1.7340
Epoch: 2/50
100%|██████████| 14/14 [00:01<00:00,  7.52it/s]
100%|██████████| 4/4 [00:00<00:00,  9.95it/s]
Training Loss --> 0.8702 Val Loss --> 0.9316
Epoch: 3/50
100%|██████████| 14/14 [00:01<00:00,  7.55it/s]
100%|██████████| 4/4 [00:00<00:00, 10.60it/s]
Training Loss --> 0.6447 Val Loss --> 0.9230
Epoch: 4/50
```

```
100%|██████████| 14/14 [00:01<00:00,  7.41it/s]
100%|██████████| 4/4 [00:00<00:00,  9.13it/s]
Training Loss --> 0.7093 Val Loss --> 0.7911
Epoch: 5/50
100%|██████████| 14/14 [00:01<00:00,  7.48it/s]
100%|██████████| 4/4 [00:00<00:00, 10.49it/s]
Training Loss --> 0.5827 Val Loss --> 0.8448
Epoch: 6/50
100%|██████████| 14/14 [00:01<00:00,  7.46it/s]
100%|██████████| 4/4 [00:00<00:00, 10.27it/s]
Training Loss --> 0.5635 Val Loss --> 0.6919
Epoch: 7/50
100%|██████████| 14/14 [00:01<00:00,  7.34it/s]
100%|██████████| 4/4 [00:00<00:00, 10.37it/s]
Training Loss --> 0.4980 Val Loss --> 0.6666
Epoch: 8/50
100%|██████████| 14/14 [00:02<00:00,  6.82it/s]
100%|██████████| 4/4 [00:00<00:00,  9.47it/s]
Training Loss --> 0.4301 Val Loss --> 0.6629
Epoch: 9/50
100%|██████████| 14/14 [00:01<00:00,  7.08it/s]
100%|██████████| 4/4 [00:00<00:00,  9.39it/s]
Training Loss --> 0.6560 Val Loss --> 0.6791
Epoch: 10/50
100%|██████████| 14/14 [00:01<00:00,  7.06it/s]
100%|██████████| 4/4 [00:00<00:00,  9.63it/s]
Training Loss --> 0.5117 Val Loss --> 0.6099
Epoch: 11/50
100%|██████████| 14/14 [00:01<00:00,  7.55it/s]
100%|██████████| 4/4 [00:00<00:00,  9.87it/s]
Training Loss --> 0.5104 Val Loss --> 0.5874
Epoch: 12/50
100%|██████████| 14/14 [00:01<00:00,  7.56it/s]
100%|██████████| 4/4 [00:00<00:00, 10.43it/s]
Training Loss --> 0.4569 Val Loss --> 0.6226
Epoch: 13/50
100%|██████████| 14/14 [00:01<00:00,  7.37it/s]
100%|██████████| 4/4 [00:00<00:00,  9.69it/s]
Training Loss --> 0.3927 Val Loss --> 0.6536
Epoch: 14/50
100%|██████████| 14/14 [00:01<00:00,  7.47it/s]
100%|██████████| 4/4 [00:00<00:00, 10.32it/s]
Training Loss --> 0.4441 Val Loss --> 0.6948
Epoch 00014: reducing learning rate of group 0 to 1.0000e-04.
Epoch: 15/50
100%|██████████| 14/14 [00:01<00:00,  7.07it/s]
100%|██████████| 4/4 [00:00<00:00,  9.69it/s]
Training Loss --> 0.4262 Val Loss --> 0.6369
Epoch: 16/50
100%|██████████| 14/14 [00:01<00:00,  7.38it/s]
100%|██████████| 4/4 [00:00<00:00,  9.97it/s]
```

```
Training Loss --> 0.3490 Val Loss --> 0.5908
Epoch: 17/50
100%|██████████| 14/14 [00:01<00:00,  7.18it/s]
100%|██████████| 4/4 [00:00<00:00, 10.49it/s]
Training Loss --> 0.3224 Val Loss --> 0.5766
Epoch: 18/50
100%|██████████| 14/14 [00:01<00:00,  7.57it/s]
100%|██████████| 4/4 [00:00<00:00, 10.16it/s]
Training Loss --> 0.2892 Val Loss --> 0.5556
Epoch: 19/50
100%|██████████| 14/14 [00:01<00:00,  7.43it/s]
100%|██████████| 4/4 [00:00<00:00, 10.61it/s]
Training Loss --> 0.3431 Val Loss --> 0.5901
Epoch: 20/50
100%|██████████| 14/14 [00:01<00:00,  7.30it/s]
100%|██████████| 4/4 [00:00<00:00, 10.46it/s]
Training Loss --> 0.2948 Val Loss --> 0.6074
Epoch: 21/50
100%|██████████| 14/14 [00:01<00:00,  7.24it/s]
100%|██████████| 4/4 [00:00<00:00,  8.78it/s]
Training Loss --> 0.2834 Val Loss --> 0.5682
Epoch 00021: reducing learning rate of group 0 to 1.0000e-05.
Epoch: 22/50
100%|██████████| 14/14 [00:01<00:00,  7.25it/s]
100%|██████████| 4/4 [00:00<00:00, 10.64it/s]
Training Loss --> 0.2526 Val Loss --> 0.6216
Epoch: 23/50
100%|██████████| 14/14 [00:01<00:00,  7.26it/s]
100%|██████████| 4/4 [00:00<00:00,  7.93it/s]
Training Loss --> 0.2663 Val Loss --> 0.5336
Epoch: 24/50
100%|██████████| 14/14 [00:02<00:00,  6.16it/s]
100%|██████████| 4/4 [00:00<00:00, 10.13it/s]
Training Loss --> 0.2557 Val Loss --> 0.5680
Epoch: 25/50
100%|██████████| 14/14 [00:02<00:00,  5.63it/s]
100%|██████████| 4/4 [00:00<00:00,  9.74it/s]
Training Loss --> 0.2583 Val Loss --> 0.5866
Epoch: 26/50
100%|██████████| 14/14 [00:01<00:00,  7.35it/s]
100%|██████████| 4/4 [00:00<00:00,  9.90it/s]
Training Loss --> 0.2978 Val Loss --> 0.5543
Epoch 00026: reducing learning rate of group 0 to 1.0000e-06.
Epoch: 27/50
100%|██████████| 14/14 [00:01<00:00,  7.34it/s]
100%|██████████| 4/4 [00:00<00:00,  9.96it/s]
Training Loss --> 0.2328 Val Loss --> 0.5678
Epoch: 28/50
100%|██████████| 14/14 [00:01<00:00,  7.51it/s]
100%|██████████| 4/4 [00:00<00:00, 10.28it/s]
Training Loss --> 0.2324 Val Loss --> 0.5388
Epoch: 29/50
```

```
100%|██████████| 14/14 [00:02<00:00,  5.38it/s]
100%|██████████| 4/4 [00:00<00:00,  5.05it/s]
Training Loss --> 0.2301 Val Loss --> 0.5724
Epoch 00029: reducing learning rate of group 0 to 1.0000e-07.
Epoch: 30/50
100%|██████████| 14/14 [00:02<00:00,  5.11it/s]
100%|██████████| 4/4 [00:00<00:00,  6.96it/s]
Training Loss --> 0.2538 Val Loss --> 0.5545
Epoch: 31/50
100%|██████████| 14/14 [00:02<00:00,  4.85it/s]
100%|██████████| 4/4 [00:00<00:00,  9.33it/s]
Training Loss --> 0.2310 Val Loss --> 0.5974
Epoch: 32/50
100%|██████████| 14/14 [00:01<00:00,  7.49it/s]
100%|██████████| 4/4 [00:00<00:00,  9.77it/s]
Training Loss --> 0.2499 Val Loss --> 0.5914
Epoch 00032: reducing learning rate of group 0 to 1.0000e-08.
Epoch: 33/50
100%|██████████| 14/14 [00:02<00:00,  6.85it/s]
100%|██████████| 4/4 [00:00<00:00, 10.14it/s]
Training Loss --> 0.2955 Val Loss --> 0.5528
Early stopping at epoch 33
Returning Best Model Weights
```

In [22]:
```
model_result["history"].plot(figsize=(15, 6),
                             title="Loss per epoch",
                             xlabel="Epoch", ylabel="Loss")
plt.show()
```



In [27]:
```
img_path_1 = "/kaggle/input/test-image-file/test_image.png"
img_path_2 = "/kaggle/input/test-image-file/random_image.png"
img_path_3 = "/kaggle/input/number-plate-detection/TEST/TEST.jpeg"
```

In [24]:
```
def ShowModelOutput(image_path, model_results, scaler):
    test_image = np.array(Image.open(image_path).convert("RGB"))
    test_aug = A.Compose([A.Resize(height=224, width=224), A.Normalize()])
    tst_image = torch.tensor(test_aug(image=test_image)["image"],
```

```
                            dtype=torch.float32).permute(2, 0, 1).unsqueeze(0).to("cud

    trained_model = model_results["model"]

    trained_model.eval()
    # Make predictions
    with torch.no_grad():
        output = trained_model(tst_image)

    # Reshape the output to match the expected input shape of the scaler
    output_np = output.cpu().numpy()

    # Apply inverse transform
    scaled_output = scaler.inverse_transform(output_np).astype(np.int32).flatten()

    pred_xmin, pred_xmax, pred_ymin, pred_ymax = scaled_output

    print("Model output:", [pred_xmin, pred_xmax, pred_ymin, pred_ymax])

    roi = test_image[pred_ymin:pred_ymax, pred_xmin:pred_xmax]

    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 10))

    ax[0].imshow(test_image)
    ax[0].set_title("Test car")
    ax[0].axis("off")

    ax[1].imshow(roi)
    ax[1].set_title("Extracted Plate")
    ax[1].axis("off")

    plt.show()
```

In [25]:
```
ShowModelOutput(img_path_1, model_result, scaler)
```

```
Model output: [193, 371, 328, 371]
```

In [26]:
```
ShowModelOutput(img_path_2, model_result, scaler)
```

```
Model output: [417, 567, 344, 394]
```

In [27]:
```
ShowModelOutput(img_path_3, model_result, scaler)
```

```
Model output: [347, 518, 393, 433]
```

Observation:

- The suggested reason for the bad prediction is due to insufficient data.
- One of the reason the model isn't performing too good may be cause of the way i scaled the target.

In [28]:
```
XTRAIN, XVAL = train_test_split(df, test_size=.2, random_state=True)

print(XTRAIN.shape, XVAL.shape)

XTRAIN.reset_index(drop=True, inplace=True)
XVAL.reset_index(drop=True, inplace=True)
```

```
(180, 7) (45, 7)
```

In [29]:
```
XTRAIN
```

Out[29]:

	imagepath	xmin	xmax	ymin	ymax	width	height
0	/kaggle/input/number-plate-detection/images/N2...	157	290	193	229	700	466
1	/kaggle/input/number-plate-detection/images/N2...	626	768	280	359	1000	562
2	/kaggle/input/number-plate-detection/images/N1...	572	839	461	535	1024	768
3	/kaggle/input/number-plate-detection/images/N1...	11	246	78	165	250	250
4	/kaggle/input/number-plate-detection/images/N1...	184	342	220	257	484	456
...	...	...	...	...	...	...	...
175	/kaggle/input/number-plate-detection/images/N2...	121	215	120	147	317	215
176	/kaggle/input/number-plate-detection/images/N1...	66	154	166	197	377	244
177	/kaggle/input/number-plate-detection/images/N1...	582	853	622	691	1920	1080
178	/kaggle/input/number-plate-detection/images/N1...	186	367	137	232	600	450
179	/kaggle/input/number-plate-detection/images/N2...	202	386	190	227	600	337

180 rows × 7 columns

In [30]:

```
# data_augment = A.Compose([
#                             A.Resize(height=224, width=224, always_apply=True),
#                             A.Normalize(always_apply=True),
#                             A.HorizontalFlip(p=0.5),
#                             A.RandomBrightnessContrast(p=0.2),
#                             A.ShiftScaleRotate(shift_limit=0.1, scale_limit=0.1,
#                             A.HueSaturationValue(p=0.3),
#                             A.RGBShift(p=0.1),
#                             A.RandomGamma(p=0.1),
#                             A.Blur(blur_limit=3, p=0.1),
#                             A.CoarseDropout(p=0.1)
#                         ])

TRAIN_DATA = ImageData(df=XTRAIN, norm_label=True)
VAL_DATA = ImageData(df=XVAL, norm_label=True)

TRAIN_DATALOADER = DataLoader(TRAIN_DATA, batch_size=13, shuffle=True)
VAL_DATALOADER = DataLoader(VAL_DATA, batch_size=13, shuffle=True)

print("Data Loader successfully created")
```

```
Data Loader successfully created
```

In [31]:
```python
base_model = mobilenet_v2(pretrained=True)

for param in base_model.parameters():
    if isinstance(param, nn.BatchNorm2d):
        param.requires_grad = False

base_model.classifier = nn.Sequential(
                            nn.Dropout(p=0.2, inplace=False),
                            nn.Linear(in_features=1280, out_features=4, bias=True)
                            )

base_model
```

Out[31]:
```
MobileNetV2(
  (features): Sequential(
    (0): Conv2dNormActivation(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=F
alse)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), g
roups=32, bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), g
roups=96, bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (3): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
```

```
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
```

```
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (12): InvertedResidual(
```

```
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (13): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (14): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
groups=576, bias=False)
          (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
```

```
stats=True)
      )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2dNormActivation(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (18): Conv2dNormActivation(
      (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      (2): ReLU6(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=4, bias=True)
  )
)
```

In [32]:
```
loss_func = RMSELoss()
optimizers = torch.optim.Adam(params=base_model.parameters(), lr=0.001)
scheduler = ReduceLROnPlateau(optimizers, mode='min', factor=0.1, patience=2, verbo
early_stopping = EarlyStopping(patience=10, min_delta=0.01, restore_best_weights=Tr

trainer = DataTrainer(base_model,
                      dataloader=[TRAIN_DATALOADER, VAL_DATALOADER],
                      optimizer=optimizers,
                      loss_function=loss_func,
                      lr_scheduler=scheduler,
                      early_stopping=early_stopping)
```

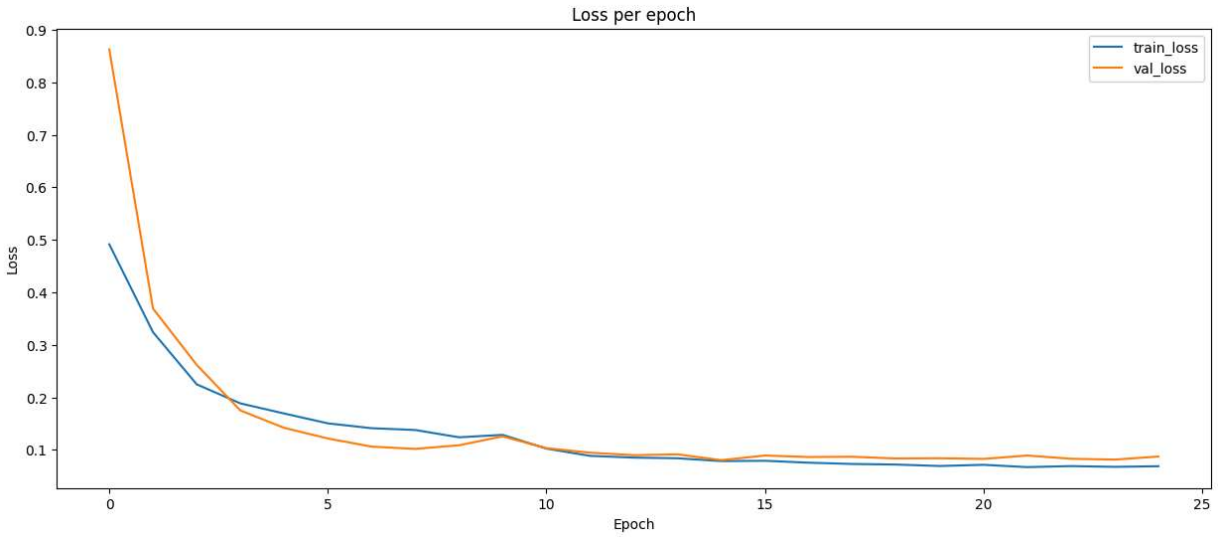
In [33]:
```
# Fit the model
model_result = trainer.fit(epochs=50)
```

```
Epoch: 1/50
100%|██████████| 14/14 [00:02<00:00,  5.25it/s]
100%|██████████| 4/4 [00:00<00:00,  5.17it/s]
Training Loss --> 0.4914 Val Loss --> 0.8625
Epoch: 2/50
100%|██████████| 14/14 [00:02<00:00,  5.21it/s]
100%|██████████| 4/4 [00:00<00:00,  5.89it/s]
Training Loss --> 0.3241 Val Loss --> 0.3691
Epoch: 3/50
100%|██████████| 14/14 [00:03<00:00,  4.35it/s]
100%|██████████| 4/4 [00:00<00:00,  5.73it/s]
Training Loss --> 0.2248 Val Loss --> 0.2619
Epoch: 4/50
100%|██████████| 14/14 [00:03<00:00,  4.50it/s]
100%|██████████| 4/4 [00:00<00:00,  6.40it/s]
Training Loss --> 0.1885 Val Loss --> 0.1749
Epoch: 5/50
100%|██████████| 14/14 [00:02<00:00,  5.49it/s]
100%|██████████| 4/4 [00:00<00:00,  9.75it/s]
```

```
Training Loss --> 0.1693 Val Loss --> 0.1420
Epoch: 6/50
100%|██████████| 14/14 [00:01<00:00,  7.51it/s]
100%|██████████| 4/4 [00:00<00:00, 10.56it/s]
Training Loss --> 0.1505 Val Loss --> 0.1217
Epoch: 7/50
100%|██████████| 14/14 [00:01<00:00,  7.61it/s]
100%|██████████| 4/4 [00:00<00:00, 10.28it/s]
Training Loss --> 0.1413 Val Loss --> 0.1062
Epoch: 8/50
100%|██████████| 14/14 [00:01<00:00,  7.57it/s]
100%|██████████| 4/4 [00:00<00:00, 10.41it/s]
Training Loss --> 0.1378 Val Loss --> 0.1019
Epoch: 9/50
100%|██████████| 14/14 [00:02<00:00,  4.88it/s]
100%|██████████| 4/4 [00:00<00:00,  6.23it/s]
Training Loss --> 0.1240 Val Loss --> 0.1088
Epoch: 10/50
100%|██████████| 14/14 [00:03<00:00,  4.56it/s]
100%|██████████| 4/4 [00:00<00:00,  5.75it/s]
Training Loss --> 0.1286 Val Loss --> 0.1257
Epoch: 11/50
100%|██████████| 14/14 [00:03<00:00,  4.57it/s]
100%|██████████| 4/4 [00:00<00:00,  5.20it/s]
Training Loss --> 0.1024 Val Loss --> 0.1032
Epoch 00011: reducing learning rate of group 0 to 1.0000e-04.
Epoch: 12/50
100%|██████████| 14/14 [00:02<00:00,  5.69it/s]
100%|██████████| 4/4 [00:00<00:00, 10.28it/s]
Training Loss --> 0.0886 Val Loss --> 0.0947
Epoch: 13/50
100%|██████████| 14/14 [00:02<00:00,  6.11it/s]
100%|██████████| 4/4 [00:00<00:00,  6.80it/s]
Training Loss --> 0.0853 Val Loss --> 0.0899
Epoch: 14/50
100%|██████████| 14/14 [00:01<00:00,  7.59it/s]
100%|██████████| 4/4 [00:00<00:00, 10.49it/s]
Training Loss --> 0.0840 Val Loss --> 0.0915
Epoch: 15/50
100%|██████████| 14/14 [00:01<00:00,  7.45it/s]
100%|██████████| 4/4 [00:00<00:00,  9.89it/s]
Training Loss --> 0.0786 Val Loss --> 0.0804
Epoch: 16/50
100%|██████████| 14/14 [00:01<00:00,  7.50it/s]
100%|██████████| 4/4 [00:00<00:00, 10.21it/s]
Training Loss --> 0.0792 Val Loss --> 0.0893
Epoch: 17/50
100%|██████████| 14/14 [00:02<00:00,  5.00it/s]
100%|██████████| 4/4 [00:00<00:00,  5.16it/s]
Training Loss --> 0.0756 Val Loss --> 0.0864
Epoch: 18/50
```

```
100%|██████████| 14/14 [00:02<00:00,  4.95it/s]
100%|██████████| 4/4 [00:00<00:00,  4.75it/s]
Training Loss --> 0.0732 Val Loss --> 0.0869
Epoch 00018: reducing learning rate of group 0 to 1.0000e-05.
Epoch: 19/50
100%|██████████| 14/14 [00:03<00:00,  3.73it/s]
100%|██████████| 4/4 [00:00<00:00,  4.26it/s]
Training Loss --> 0.0721 Val Loss --> 0.0836
Epoch: 20/50
100%|██████████| 14/14 [00:02<00:00,  6.10it/s]
100%|██████████| 4/4 [00:00<00:00, 10.04it/s]
Training Loss --> 0.0693 Val Loss --> 0.0841
Epoch: 21/50
100%|██████████| 14/14 [00:01<00:00,  7.47it/s]
100%|██████████| 4/4 [00:00<00:00, 10.16it/s]
Training Loss --> 0.0716 Val Loss --> 0.0828
Epoch 00021: reducing learning rate of group 0 to 1.0000e-06.
Epoch: 22/50
100%|██████████| 14/14 [00:01<00:00,  7.26it/s]
100%|██████████| 4/4 [00:00<00:00, 10.20it/s]
Training Loss --> 0.0671 Val Loss --> 0.0892
Epoch: 23/50
100%|██████████| 14/14 [00:01<00:00,  7.30it/s]
100%|██████████| 4/4 [00:00<00:00, 10.54it/s]
Training Loss --> 0.0691 Val Loss --> 0.0831
Epoch: 24/50
100%|██████████| 14/14 [00:01<00:00,  7.34it/s]
100%|██████████| 4/4 [00:00<00:00,  9.69it/s]
Training Loss --> 0.0677 Val Loss --> 0.0815
Epoch 00024: reducing learning rate of group 0 to 1.0000e-07.
Epoch: 25/50
100%|██████████| 14/14 [00:01<00:00,  7.31it/s]
100%|██████████| 4/4 [00:00<00:00,  9.74it/s]
Training Loss --> 0.0688 Val Loss --> 0.0873
Early stopping at epoch 25
Returning Best Model Weights
```

In [34]:
```
model_result["history"].plot(figsize=(15, 6),
                             title="Loss per epoch",
                             xlabel="Epoch", ylabel="Loss")
plt.show()
```



In [45]:
```
def ShowModelOutput_2(image_path, model_results):
    test_image = np.array(Image.open(image_path).convert("RGB"))
    h, w, c = test_image.shape
    test_aug = A.Compose([A.Resize(height=224, width=224), A.Normalize()])
    tst_image = torch.tensor(test_aug(image=test_image)["image"],
                             dtype=torch.float32).permute(2, 0, 1).unsqueeze(0).to("cud

    trained_model = model_results["model"]

    trained_model.eval()
    # Make predictions
    with torch.no_grad():
        output = trained_model(tst_image)

    # Reshape the output to match the expected input shape of the scaler
    output_np = output.cpu().numpy()

    denorm = np.array([w,w,h,h])

    pred_xmin, pred_xmax, pred_ymin, pred_ymax = (output_np * denorm).astype(np.int

    print("Model output:", [pred_xmin, pred_xmax, pred_ymin, pred_ymax])

    roi = test_image[pred_ymin:pred_ymax, pred_xmin:pred_xmax]

    fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 10))

    ax[0].imshow(test_image)
    ax[0].set_title("Test car")
    ax[0].axis("off")

    ax[1].imshow(roi)
    ax[1].set_title(f"Extracted Plate -> {pytesseract.image_to_string(roi)}")
    ax[1].axis("off")

    plt.show()
```

In [46]:
```
ShowModelOutput_2(img_path_1, model_result)
```

```
Model output: [292, 520, 404, 449]
```

Test car

Extracted Plate -> IVMGS |
□

In [47]:
```
ShowModelOutput_2(img_path_2, model_result)
```

```
Model output: [459, 642, 408, 445]
```

Test car

Extracted Plate -> □

In [49]:
```
ShowModelOutput_2(img_path_3, model_result)
```

```
Model output: [374, 574, 477, 534]
```

Test car

Extracted Plate -> Si K 4999
□

Observation:

- The model performed better than last time but not 100% accurate.
- I would explore a more robust model that's capable of performing well even on small dataset.

Modelling with YoloV9

In [7]:
```
annotated_filepath = "/kaggle/input/car-plate-detection/annotations"
image_filepath = "/kaggle/input/car-plate-detection/images"
def LoadDf(annotated_filepath:str, image_filepath:str):
    data = {"carpath":[], "xmin":[], "xmax":[], "ymin":[], "ymax":[], "width":[], "
    for i in glb(annotated_filepath+"/*.xml"):
        root_element = ElementTree.parse(i).getroot()
        car_path = os.path.join(image_filepath, root_element.find("filename").text)
        bounding_box = root_element.find("object").find("bndbox")
        size = root_element.find("size")
        data["carpath"].append(car_path)
        data["xmin"].append(int(bounding_box.find("xmin").text))
        data["xmax"].append(int(bounding_box.find("xmax").text))
        data["ymin"].append(int(bounding_box.find("ymin").text))
        data["ymax"].append(int(bounding_box.find("ymax").text))
        data["width"].append(int(size.find("width").text))
        data["height"].append(int(size.find("height").text))
    return pd.DataFrame(data)
```

In [8]:
```
df_1 = LoadDf(annotated_filepath, image_filepath)

print(df_1.shape)

df_1.head()
```

```
(433, 7)
```

Out[8]:

	carpath	xmin	xmax	ymin	ymax	width	height
0	/kaggle/input/car-plate-detection/images/Cars3...	209	283	135	169	500	300
1	/kaggle/input/car-plate-detection/images/Cars1...	191	242	147	169	400	268
2	/kaggle/input/car-plate-detection/images/Cars7...	115	277	115	153	400	267
3	/kaggle/input/car-plate-detection/images/Cars1...	36	62	175	186	400	221
4	/kaggle/input/car-plate-detection/images/Cars2...	71	215	205	246	517	303

In [9]:
```
def LoadData(image_path):
    data = {"carpath":[], "xmin":[], "xmax":[], "ymin":[], "ymax":[], "width":[], "
    file_path = glb(image_path+"/*.xml")
    for xml_path in file_path:
        root_element = ElementTree.parse(xml_path).getroot()
```

```
        imagepath = os.path.join(image_path, root_element.find("filename").text)
        bounding_box = root_element.find("object").find("bndbox")
        size = root_element.find("size")
        data["carpath"].append(imagepath)
        data["xmin"].append(int(bounding_box.find("xmin").text))
        data["xmax"].append(int(bounding_box.find("xmax").text))
        data["ymin"].append(int(bounding_box.find("ymin").text))
        data["ymax"].append(int(bounding_box.find("ymax").text))
        data["width"].append(int(size.find("width").text))
        data["height"].append(int(size.find("height").text))
    return pd.DataFrame(data)
```

In [10]:
```
image_path = "/kaggle/input/number-plate-detection/images"

df_2 = LoadData(image_path)

print(df_2.shape)

df_2.head()
```

```
(225, 7)
```

Out[10]:

	carpath	xmin	xmax	ymin	ymax	width	height
0	/kaggle/input/number-plate-detection/images/N1...	244	369	240	293	600	430
1	/kaggle/input/number-plate-detection/images/N1...	331	538	263	317	850	681
2	/kaggle/input/number-plate-detection/images/N1...	80	335	150	243	445	349
3	/kaggle/input/number-plate-detection/images/N2...	131	209	129	153	350	196
4	/kaggle/input/number-plate-detection/images/N1...	180	559	216	314	678	475

In [11]:
```
combined_df = pd.concat([df_1, df_2], axis="rows")

print(combined_df.shape)

combined_df.head()
```

```
(658, 7)
```

Out[11]:

	carpath	xmin	xmax	ymin	ymax	width	height
0	/kaggle/input/car-plate-detection/images/Cars3...	209	283	135	169	500	300
1	/kaggle/input/car-plate-detection/images/Cars1...	191	242	147	169	400	268
2	/kaggle/input/car-plate-detection/images/Cars7...	115	277	115	153	400	267
3	/kaggle/input/car-plate-detection/images/Cars1...	36	62	175	186	400	221
4	/kaggle/input/car-plate-detection/images/Cars2...	71	215	205	246	517	303

In [12]:
```
combined_df.isnull().sum()
```

Out[12]:
```
carpath    0
xmin       0
xmax       0
ymin       0
ymax       0
width      0
height     0
dtype: int64
```

In [13]:
```
combined_df['center_x'] = (combined_df['xmax'] + combined_df['xmin'])/(2*combined_d
combined_df['center_y'] = (combined_df['ymax'] + combined_df['ymin'])/(2*combined_d

combined_df['bb_width'] = (combined_df['xmax'] - combined_df['xmin'])/combined_df['
combined_df['bb_height'] = (combined_df['ymax'] - combined_df['ymin'])/combined_df[

combined_df.head()
```

Out[13]:

	carpath	xmin	xmax	ymin	ymax	width	height	center_x	center_y	I
0	/kaggle/input/car-plate-detection/images/Cars3...	209	283	135	169	500	300	0.492000	0.506667	
1	/kaggle/input/car-plate-detection/images/Cars1...	191	242	147	169	400	268	0.541250	0.589552	
2	/kaggle/input/car-plate-detection/images/Cars7...	115	277	115	153	400	267	0.490000	0.501873	
3	/kaggle/input/car-plate-detection/images/Cars1...	36	62	175	186	400	221	0.122500	0.816742	
4	/kaggle/input/car-plate-detection/images/Cars2...	71	215	205	246	517	303	0.276596	0.744224	

In [14]:
```
train_x,val_x = train_test_split(combined_df, test_size=.1)

print(train_x.shape, val_x.shape)
```

```
(592, 11) (66, 11)
```

In [15]:
```
train_x
```

Out[15]:

	carpath	xmin	xmax	ymin	ymax	width	height	center_x	center_y
60	/kaggle/input/car-plate-detection/images/Cars8...	251	326	295	341	600	450	0.480833	0.706667
131	/kaggle/input/car-plate-detection/images/Cars1...	159	250	178	200	400	265	0.511250	0.713208
30	/kaggle/input/car-plate-detection/images/Cars1...	315	407	287	311	507	388	0.712032	0.770619
223	/kaggle/input/number-plate-detection/images/N5...	1238	2468	1012	1269	3720	2472	0.498118	0.461367
19	/kaggle/input/number-plate-detection/images/N9...	239	347	187	211	600	400	0.488333	0.497500
...	...	...	...	...	...	...	...	...	...
71	/kaggle/input/car-plate-detection/images/Cars4...	204	280	103	175	400	251	0.605000	0.553785
106	/kaggle/input/car-plate-detection/images/Cars7...	166	206	141	154	400	279	0.465000	0.528674
270	/kaggle/input/car-plate-detection/images/Cars4...	189	352	193	248	400	353	0.676250	0.624646
2	/kaggle/input/number-plate-detection/images/N1...	80	335	150	243	445	349	0.466292	0.563037
102	/kaggle/input/car-plate-detection/images/Cars3...	172	437	163	220	600	417	0.507500	0.459233

592 rows × 11 columns

In [16]:
```
def prepara_yaml(train_df, val_df):
    # Define the root paths for images and labels
    train_images_path = "data/train/images"
    train_labels_path = "data/train/labels"
    val_images_path = "data/val/images"
    val_labels_path = "data/val/labels"

    # Create directories if they don't exist
    os.makedirs(train_images_path, exist_ok=True)
```

```
    os.makedirs(train_labels_path, exist_ok=True)
    os.makedirs(val_images_path, exist_ok=True)
    os.makedirs(val_labels_path, exist_ok=True)

    # Copy training images and annotations
    for index, row in train_df.iterrows():
        # Copy image
        shutil.copy(src=row["carpath"], dst=os.path.join(train_images_path, os.path

        # Create label file
        label_path = os.path.join(train_labels_path, f'{os.path.basename(row["carpa
        with open(label_path, 'w') as file:
            file.write(f'0 {row["center_x"]:.4f} {row["center_y"]:.4f} {row["bb_wid

    # Copy validation images and annotations
    for index, row in val_df.iterrows():
        # Copy image
        shutil.copy(src=row["carpath"], dst=os.path.join(val_images_path, os.path.b

        # Create label file
        label_path = os.path.join(val_labels_path, f'{os.path.basename(row["carpath
        with open(label_path, 'w') as file:
            file.write(f'0 {row["center_x"]:.4f} {row["center_y"]:.4f} {row["bb_wid

    print("Done")
```

In [17]:
```
prepara_yaml(train_x,val_x)
```

```
Done
```

In [18]:
```
data = {
    "train": "/kaggle/working/data/train/images",
    "val": "/kaggle/working/data/val/images",
    "nc":1,
    "name": ["license_plate"]
}

with open("data.yaml", "w+") as file:
    yaml.dump(data, file)
```

In [19]:
```
yolo_model = YOLO("yolov9c.pt")
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov9c.p
t to 'yolov9c.pt'...
100%|████████| 49.4M/49.4M [00:00<00:00, 259MB/s]
```

In [20]:
```
results = yolo_model.train(data='/kaggle/working/data.yaml',
                           epochs=30,
#                            imgsz = 416,
                           batch = 15,
                                 # lr=0.0001,
#                            dropout= 0.5,
                           device = 0)
```

```
Ultralytics YOLOv8.2.28 🚀 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla P100-PCIE-16GB, 16276MiB)
engine/trainer: task=detect, mode=train, model=yolov9c.pt, data=/kaggle/working/data.yaml, epochs=30, time=None, patience=100, batch=15, imgsz=640, save=True, save_period=-1, cache=False, device=0, workers=8, project=None, name=train, exist_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single_cls=False, rect=False, cos_lr=False, close_mosaic=10, resume=False, amp=True, fraction=1.0, profile=False, freeze=None, multi_scale=False, overlap_mask=True, mask_ratio=4, dropout=0.0, val=True, split=val, save_json=False, save_hybrid=False, conf=None, iou=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, vid_stride=1, stream_buffer=False, visualize=False, augment=False, agnostic_nms=False, classes=None, retina_masks=False, embed=None, show=False, save_frames=False, save_txt=False, save_conf=False, save_crop=False, show_labels=True, show_conf=True, show_boxes=True, line_width=None, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify=False, opset=None, workspace=4, nms=False, lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label_smoothing=0.0, nbs=64, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, bgr=0.0, mosaic=1.0, mixup=0.0, copy_paste=0.0, auto_augment=randaugment, erasing=0.4, crop_fraction=1.0, cfg=None, tracker=botsort.yaml, save_dir=runs/detect/train
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 22.0MB/s]
2024-06-03 11:54:33,878 INFO util.py:124 -- Outdated packages:
  ipywidgets==7.7.1 found, needs ipywidgets>=8
Run `pip install -U ipywidgets`, then restart the notebook server for rich notebook output.
2024-06-03 11:54:34,688 INFO util.py:124 -- Outdated packages:
  ipywidgets==7.7.1 found, needs ipywidgets>=8
Run `pip install -U ipywidgets`, then restart the notebook server for rich notebook output.
```

```
Overriding model.yaml nc=80 with nc=1

                   from  n    params  module                                       a
rguments
  0                  -1  1      1856  ultralytics.nn.modules.conv.Conv
[3, 64, 3, 2]
  1                  -1  1     73984  ultralytics.nn.modules.conv.Conv
[64, 128, 3, 2]
  2                  -1  1    212864  ultralytics.nn.modules.block.RepNCSPELAN4
[128, 256, 128, 64, 1]
  3                  -1  1    164352  ultralytics.nn.modules.block.ADown
[256, 256]
  4                  -1  1    847616  ultralytics.nn.modules.block.RepNCSPELAN4
[256, 512, 256, 128, 1]
  5                  -1  1    656384  ultralytics.nn.modules.block.ADown
[512, 512]
  6                  -1  1   2857472  ultralytics.nn.modules.block.RepNCSPELAN4
[512, 512, 512, 256, 1]
  7                  -1  1    656384  ultralytics.nn.modules.block.ADown
[512, 512]
  8                  -1  1   2857472  ultralytics.nn.modules.block.RepNCSPELAN4
[512, 512, 512, 256, 1]
  9                  -1  1    656896  ultralytics.nn.modules.block.SPPELAN
[512, 512, 256]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 12                  -1  1   3119616  ultralytics.nn.modules.block.RepNCSPELAN4
[1024, 512, 512, 256, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 15                  -1  1    912640  ultralytics.nn.modules.block.RepNCSPELAN4
[1024, 256, 256, 128, 1]
 16                  -1  1    164352  ultralytics.nn.modules.block.ADown
[256, 256]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 18                  -1  1   2988544  ultralytics.nn.modules.block.RepNCSPELAN4
[768, 512, 512, 256, 1]
 19                  -1  1    656384  ultralytics.nn.modules.block.ADown
[512, 512]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat
[1]
 21                  -1  1   3119616  ultralytics.nn.modules.block.RepNCSPELAN4
[1024, 512, 512, 256, 1]
 22        [15, 18, 21]  1   5583571  ultralytics.nn.modules.head.Detect
[1, [256, 512, 512]]
YOLOv9c summary: 618 layers, 25530003 parameters, 25529987 gradients, 103.7 GFLOPs

Transferred 931/937 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/train', view at http://loc
alhost:6006/
```

```
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit:
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
```

Tracking run with wandb version 0.17.0

Run data is saved locally in /kaggle/working/wandb/run-20240603_115545-2tuszbrq

Syncing run **train** to Weights & Biases (docs)

View project at https://wandb.ai/eddyblizzy14/YOLOv8

View run at https://wandb.ai/eddyblizzy14/YOLOv8/runs/2tuszbrq

```
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8n.pt to 'yolov8n.pt'...
100%|██████████| 6.23M/6.23M [00:00<00:00, 129MB/s]
AMP: checks passed ✅
train: Scanning /kaggle/working/data/train/labels... 592 images, 0 backgrounds, 0 corrupt: 100%|██████████| 592/592 [00:00<00:00, 627.18it/s]
train: New cache created: /kaggle/working/data/train/labels.cache
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
val: Scanning /kaggle/working/data/val/labels... 66 images, 0 backgrounds, 0 corrupt: 100%|██████████| 66/66 [00:00<00:00, 712.51it/s]
val: New cache created: /kaggle/working/data/val/labels.cache
Plotting labels to runs/detect/train/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 154 weight(decay=0.0), 161 weight(decay=0.00046875), 160 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 640 train, 640 val
Using 4 dataloader workers
Logging results to runs/detect/train
Starting training for 30 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       1/30      11.1G      1.566      3.036      1.395         14        640: 100%|██████████| 40/40 [00:36<00:00,  1.09it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 3/3 [00:02<00:00,  1.33it/s]
                   all         66         66     0.0169      0.409     0.0118    0.00463

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       2/30      10.9G      1.633      1.544      1.443         17        640: 100%|██████████| 40/40 [00:34<00:00,  1.18it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 3/3 [00:01<00:00,  2.82it/s]
                   all         66         66   0.000663      0.197   0.000428   0.00132

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
```

```
       3/30      10.9G      1.671      1.466      1.519         11        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.58it/s]
                   all         66         66   0.000678      0.167   0.000406   0.00
0128

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/30      10.9G      1.652      1.402      1.496         13        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.91it/s]
                   all         66         66    0.00051     0.0152   0.000139   4.18
e-05

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       5/30      10.9G      1.624       1.27      1.503         10        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.87it/s]
                   all         66         66      0.448      0.303      0.278
0.114

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/30      10.9G      1.545      1.195      1.426         11        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.448      0.303      0.278
0.114

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/30      10.9G      1.546       1.12      1.425         13        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.86it/s]
                   all         66         66      0.732      0.591      0.651
0.316

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       8/30      10.9G       1.46      1.061      1.374         14        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.80it/s]
                   all         66         66      0.745      0.398      0.503
0.257

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       9/30      10.9G       1.46      1.061      1.373          9        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.88it/s]
                   all         66         66      0.704      0.755      0.767
0.397

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
```

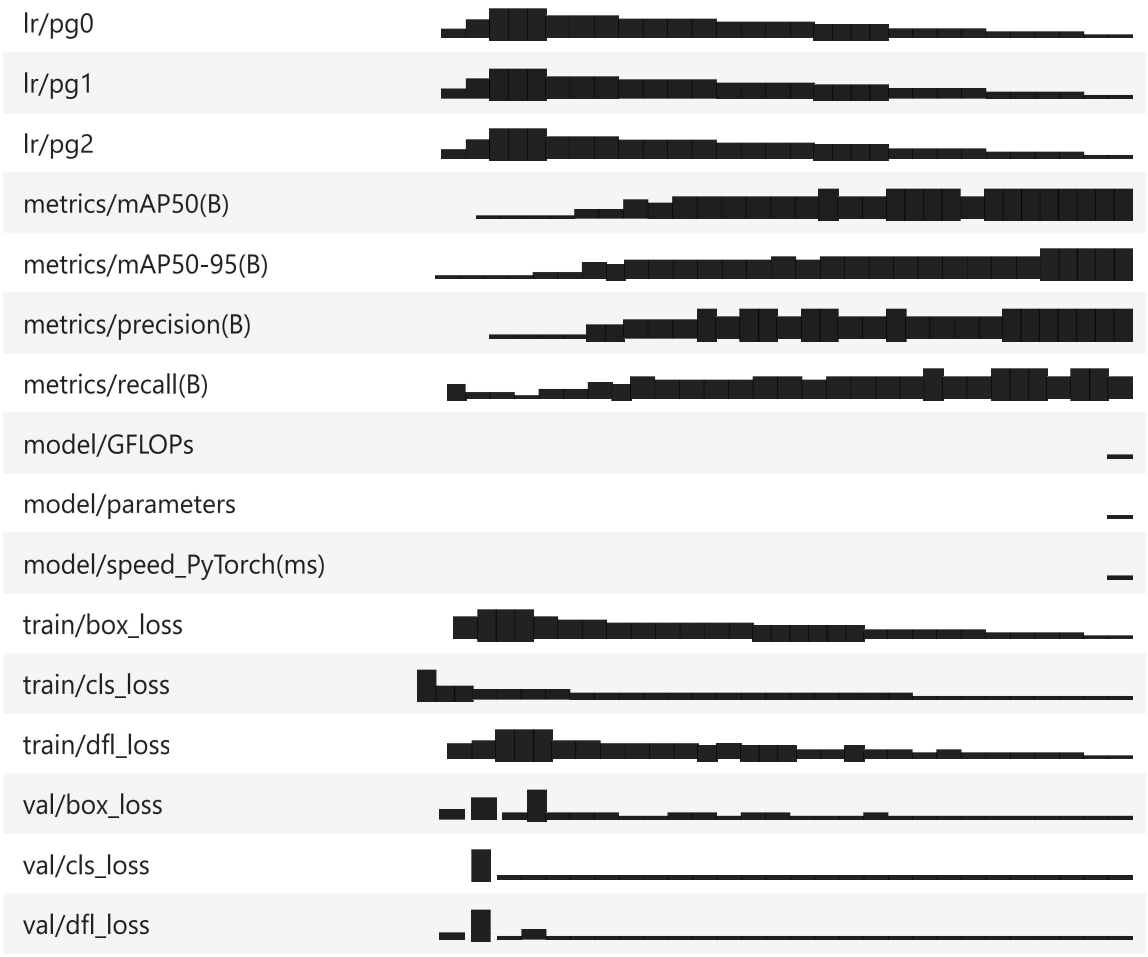
```
      10/30      10.9G      1.452      1.054      1.371         12        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.87it/s]
                   all         66         66      0.901      0.688      0.813
0.375

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      11/30      10.9G      1.434     0.9883      1.352         13        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.87it/s]
                   all         66         66      0.787      0.727      0.766
0.387

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      12/30      10.9G      1.418     0.9322      1.349         11        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.90it/s]
                   all         66         66      0.936      0.652      0.824
0.418

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      13/30      10.9G      1.431      0.909      1.364         15        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.903      0.706      0.844
0.417

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      14/30      10.9G      1.334     0.8678      1.315         13        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.88it/s]
                   all         66         66      0.844      0.742      0.814
0.436

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      15/30      10.9G      1.344     0.8915      1.316         10        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.88it/s]
                   all         66         66      0.947      0.815      0.908
0.463

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      16/30      10.9G      1.362     0.8333      1.302         11        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.91it/s]
                   all         66         66      0.901      0.691      0.759
0.385

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
```

```
      17/30      10.9G      1.321     0.8251      1.282         13        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.87it/s]
                   all         66         66      0.873      0.773      0.852
0.448

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      18/30      10.9G      1.322     0.8445      1.289         16        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.91it/s]
                   all         66         66      0.818      0.848      0.887
0.502

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/30      10.9G      1.301     0.8136      1.306         14        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.87it/s]
                   all         66         66      0.931      0.821      0.877
0.444

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/30      10.9G      1.264     0.7788      1.278          9        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.875      0.833      0.886
0.461

Closing dataloader mosaic
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3,
7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      21/30      10.9G      1.254     0.7712      1.283          7        640: 100%|
████████| 40/40 [00:34<00:00,  1.15it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.841      0.879      0.886
0.492

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      22/30      10.9G      1.216     0.7004      1.246          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.88it/s]
                   all         66         66      0.867      0.833       0.86
0.469

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      23/30      10.9G      1.248     0.7051      1.258          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.90it/s]
                   all         66         66      0.851      0.862      0.883
0.515
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      24/30      10.9G      1.215     0.6912      1.238          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.84it/s]
                   all         66         66      0.923      0.894      0.937
0.519
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      25/30      10.9G      1.201     0.6719      1.248          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.91it/s]
                   all         66         66      0.907      0.894        0.9
0.518
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      26/30      10.9G      1.175     0.6549      1.218          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.82it/s]
                   all         66         66      0.912      0.939      0.931
0.532
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      27/30      10.9G      1.166     0.6178      1.221          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.938      0.864      0.938
0.56
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      28/30      10.9G      1.153     0.6162      1.219          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.92it/s]
                   all         66         66      0.917      0.894      0.926
0.538
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      29/30      10.9G      1.121     0.5968      1.181          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.92it/s]
                   all         66         66      0.951      0.875      0.937
0.541
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      30/30      10.9G      1.089     0.5789      1.183          7        640: 100%|
████████| 40/40 [00:33<00:00,  1.19it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|████████| 3/3 [00:01<00:00,  2.89it/s]
                   all         66         66      0.895      0.864      0.917
0.541
```

```
30 epochs completed in 0.315 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 51.6MB
Optimizer stripped from runs/detect/train/weights/best.pt, 51.6MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.2.28 🚀 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla P100-PCIE-16GB,
16276MiB)
YOLOv9c summary (fused): 384 layers, 25320019 parameters, 0 gradients, 102.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50
-95): 100%|██████████| 3/3 [00:01<00:00,  2.20it/s]
                   all         66         66      0.941      0.864      0.938
0.559
Speed: 0.2ms preprocess, 14.4ms inference, 0.0ms loss, 1.0ms postprocess per image
Results saved to runs/detect/train
VBox(children=(Label(value='55.016 MB of 55.016 MB uploaded\r'), FloatProgress(value
=1.0, max=1.0)))
```

Run history:



Run summary:

lr/pg0	9e-05
lr/pg1	9e-05
lr/pg2	9e-05
metrics/mAP50(B)	0.93821
metrics/mAP50-95(B)	0.55877
metrics/precision(B)	0.94068
metrics/recall(B)	0.86364
model/GFLOPs	103.683
model/parameters	25530003

model/speed_PyTorch(ms)	13.113
train/box_loss	1.08871
train/cls_loss	0.57894
train/dfl_loss	1.18301
val/box_loss	1.57131
val/cls_loss	0.67818
val/dfl_loss	1.51783

View run **train** at: https://wandb.ai/eddyblizzy14/YOLOv8/runs/2tuszbrq
View project at: https://wandb.ai/eddyblizzy14/YOLOv8
Synced 5 W&B file(s), 24 media file(s), 5 artifact file(s) and 0 other file(s)
Find logs at: `./wandb/run-20240603_115545-2tuszbrq/logs`

In [85]:
```
model_path = "/kaggle/working/runs/detect/train/weights/best.pt"

def ShowModelPrediction(model_path:str, file_path:str):

    pre_trained_yolo_model = YOLO(model_path)

    prediction = pre_trained_yolo_model(file_path)

    pred_info = prediction[0].boxes

    loaded_image = np.array(Image.open(file_path).convert("RGB"))

    print(pred_info, end="\n\n")

    if len(pred_info.xyxy.to("cpu").numpy()) == 1:
        pred_xmin, pred_ymin, pred_xmax, pred_ymax = pred_info.xyxy.to("cpu").numpy

        roi = loaded_image[pred_ymin:pred_ymax, pred_xmin:pred_xmax]

        fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 10))

        ax[0].imshow(loaded_image)
        ax[0].set_title("Test car")
        ax[0].axis("off")

        ax[1].imshow(roi)
        ax[1].set_title(f"Extracted Plate -> {pytesseract.image_to_string(roi)}")
        ax[1].axis("off")

        plt.show()

    else:
        for (xmin, ymin, xmax, ymax) in pred_info.xyxy.to("cpu").numpy().astype(np.
```

```
            roi = loaded_image[ymin:ymax, xmin : xmax]

            fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 10))

            ax[0].imshow(loaded_image)
            ax[0].set_title("Test car")
            ax[0].axis("off")

            ax[1].imshow(roi)
            ax[1].set_title(f"Extracted Plate -> {pytesseract.image_to_string(roi)}")
            ax[1].axis("off")

        plt.show()
```

In [86]:
```
ShowModelPrediction(model_path, img_path_1)
```

```
image 1/1 /kaggle/input/test-image-file/test_image.png: 544x640 1 class_0, 23.5ms
Speed: 2.6ms preprocess, 23.5ms inference, 1.6ms postprocess per image at shape (1,
3, 544, 640)
ultralytics.engine.results.Boxes object with attributes:

cls: tensor([0.], device='cuda:0')
conf: tensor([0.8099], device='cuda:0')
data: tensor([[283.7371, 390.7187, 488.5140, 454.1490,   0.8099,   0.0000]], device
='cuda:0')
id: None
is_track: False
orig_shape: (628, 772)
shape: torch.Size([1, 6])
xywh: tensor([[386.1255, 422.4339, 204.7770,  63.4303]], device='cuda:0')
xywhn: tensor([[0.5002, 0.6727, 0.2653, 0.1010]], device='cuda:0')
xyxy: tensor([[283.7371, 390.7187, 488.5140, 454.1490]], device='cuda:0')
xyxyn: tensor([[0.3675, 0.6222, 0.6328, 0.7232]], device='cuda:0')
```

In [87]:
```
ShowModelPrediction(model_path, img_path_2)
```

```
image 1/1 /kaggle/input/test-image-file/random_image.png: 448x640 2 class_0s, 18.9ms
Speed: 2.3ms preprocess, 18.9ms inference, 1.4ms postprocess per image at shape (1,
3, 448, 640)
ultralytics.engine.results.Boxes object with attributes:

cls: tensor([0., 0.], device='cuda:0')
conf: tensor([0.8057, 0.5988], device='cuda:0')
data: tensor([[4.7297e+02, 3.8497e+02, 6.8301e+02, 4.6788e+02, 8.0574e-01, 0.0000e+0
0],
        [0.0000e+00, 1.6579e+02, 5.9107e+01, 1.8379e+02, 5.9877e-01, 0.0000e+00]], d
evice='cuda:0')
id: None
is_track: False
orig_shape: (570, 838)
shape: torch.Size([2, 6])
xywh: tensor([[577.9908, 426.4253, 210.0338,  82.9059],
        [ 29.5535, 174.7903,  59.1071,  18.0020]], device='cuda:0')
xywhn: tensor([[0.6897, 0.7481, 0.2506, 0.1454],
        [0.0353, 0.3066, 0.0705, 0.0316]], device='cuda:0')
xyxy: tensor([[472.9738, 384.9723, 683.0077, 467.8782],
        [  0.0000, 165.7893,  59.1071, 183.7913]], device='cuda:0')
xyxyn: tensor([[0.5644, 0.6754, 0.8150, 0.8208],
        [0.0000, 0.2909, 0.0705, 0.3224]], device='cuda:0')
```

Test car

Extracted Plate -> □

Test car

Extracted Plate -> CE6IYDM
□

In [88]:
```
ShowModelPrediction(model_path, img_path_3)
```

```
image 1/1 /kaggle/input/number-plate-detection/TEST/TEST.jpeg: 544x640 1 class_0, 2
2.7ms
Speed: 2.5ms preprocess, 22.7ms inference, 1.4ms postprocess per image at shape (1,
3, 544, 640)
ultralytics.engine.results.Boxes object with attributes:

cls: tensor([0.], device='cuda:0')
conf: tensor([0.8120], device='cuda:0')
data: tensor([[328.0302, 486.9390, 586.3840, 539.3282,   0.8120,   0.0000]], device
='cuda:0')
id: None
is_track: False
orig_shape: (729, 901)
shape: torch.Size([1, 6])
xywh: tensor([[457.2071, 513.1336, 258.3538,  52.3892]], device='cuda:0')
xywhn: tensor([[0.5074, 0.7039, 0.2867, 0.0719]], device='cuda:0')
xyxy: tensor([[328.0302, 486.9390, 586.3840, 539.3282]], device='cuda:0')
xyxyn: tensor([[0.3641, 0.6680, 0.6508, 0.7398]], device='cuda:0')
```

Test car

Extracted Plate -> KL 51 K 4999

Eddy