Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Обнаружение полосы движения на основе глубокого обучения

Студент ИУ5И-31М _____ Хуан Яовэнь
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ Ю.Е. Гапанюк
(Подпись, дата) (И.О.Фамилия)

Консультант _____ _____
(Подпись, дата) (И.О.Фамилия)

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой __ИУ5_____
(Индекс)
_____ _____
(И.О.Фамилия)
« _ » ____ 20 г.

З А Д А Н И Е
на выполнение курсового проекта

по теме ***Обнаружение полосы движения на основе глубокого обучения***

Студент группы ИУ5И-31М
Хуан Яовэнь
(Фамилия, имя, отчество)

Направленность курсового проекта (учебная, исследовательская, практическая, производственная, др.)
исследовательская
Источник тематики (кафедра, предприятие, НИР) Кафедра

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Техническое задание
Создать одноэтапные и двухэтапные модели, использовать изображения, собранные в интернете, для тестирования предварительно обученной модели и сравнить результаты тестирования.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на __18__ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
_____
_____
_____
_____

Дата выдачи задания « ___ » _____ 20_____ г.

Руководитель НИР _____ Ю.Е. Гапанюк
(Подпись, дата) (И.О.Фамилия)

Студент _____ Хуан Яовэнь
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Аннотация

С развитием технологии компьютерного оборудования также быстро развиваются вычисления компьютерного зрения для сегментации изображений путем извлечения признаков из большого количества данных. Для точного определения полосы движения в процессе вождения транспортного средства была создана модель обнаружения полосы движения на основе сети SegNet, и после завершения обучения был получен очень хороший результат тестирования.

Исследование завершено на платформе colab.

Содержание

1. Теоретическая часть

Методы обнаружения линии полосы движения в основном делятся на две категории: одна основана на традиционных методах машинного зрения, а другая — на методах глубокого обучения.

1.1 Методы, основанные на традиционном машинном зрении

1. Обнаружение краев + преобразование Хафа

Поток метода: цветное изображение в оттенках серого, размытие, обнаружение краев, преобразование Хафа.

Этот метод, как правило, позволяет обнаружить две полосы движения, по которым в настоящее время движется автомобиль, в простой сцене, а также случайную соседнюю полосу (в зависимости от угла камеры переднего обзора). Этот метод может использовать результат преобразования Хафа (наклон линии) для дальнейшей фильтрации линий левой и правой дорожки. Но в то же время этот метод также зависит от результата обнаружения края, поэтому очень важно настроить параметры (обнаружение края, преобразование Хафа) и другие приемы (выбор области интереса и т. д.).

2. Цветовой порог

Последовательность действий: преобразовать изображение в цветовое пространство (обычно HSV), установить пороговое значение для каждого канала в новом цветовом пространстве (значение, превышающее пороговое значение, равно 1, а значение, меньшее значения, равно 0), и результат получается.

Этот метод основан на выборе порога каждого канала и требует только настройки нескольких параметров порога, но надежность этого метода будет низкой, например, транспортные средства перед текущим транспортным средством могут быть все установлено на 1.

3. Преобразование перспективы

Последовательность действий: получение матрицы преобразования перспективы, преобразование перспективы, обнаружение линии дорожки (1 или 2)

Преимущество этого метода в том, что изображение, снятое камерой переднего вида, преобразуется в вид с высоты птичьего полета, и можно обнаружить несколько линий. Ключ заключается в точности матрицы преобразования перспективы (без учета преобразованного обнаружения линии дорожки). Для преобразованного вида с высоты птичьего полета линия дорожки может быть обнаружена двумя вышеуказанными способами.

В реальной сцене надежность традиционного метода действительно оставляет желать лучшего: помимо влияния освещения и соседних транспортных средств, стрелки указателей и тротуары в середине полосы также являются проблемами, с которыми такие алгоритмы трудно справиться.

1.2 Метод, основанный на глубоком обучении

SegNet — это модель семантической сегментации. Эта базовая обучаемая архитектура сегментации состоит из сети кодировщика, соответствующей сети декодера, за которой следует слой классификации по пикселям. Архитектура сети кодировщика топологически идентична сети VG16 в сверточных слоях Роль сети декодера заключается в сопоставлении карт объектов кодировщика с низким разрешением с картами объектов с полным входным разрешением для попиксельной классификации. Новинка SegNet заключается в том, как декодер повышает дискретизацию своих входных карт объектов с более низким разрешением. В частности, декодер использует индексы объединения, вычисленные на этапе максимального объединения соответствующего кодировщика, для выполнения нелинейной повышающей дискретизации.

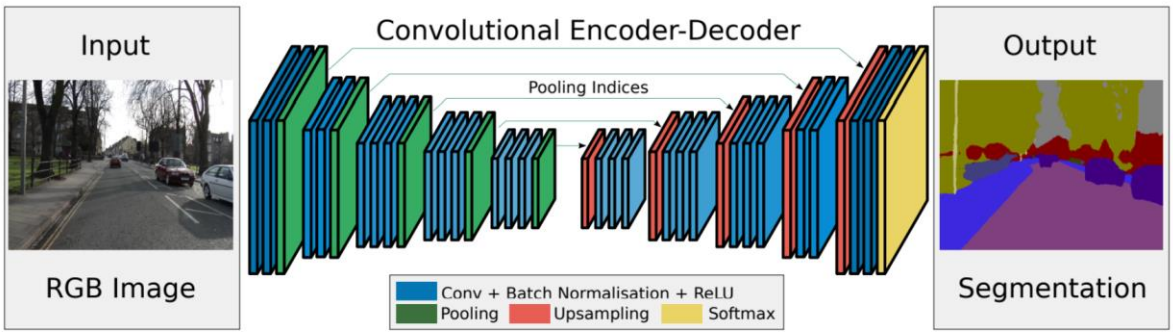


Рис 1. Структура нейронной сети SegNet

Расположение полосы движения можно получить, обучив сеть SegNet с помощью набора данных.

Изображения и метки в наборе данных следующие:

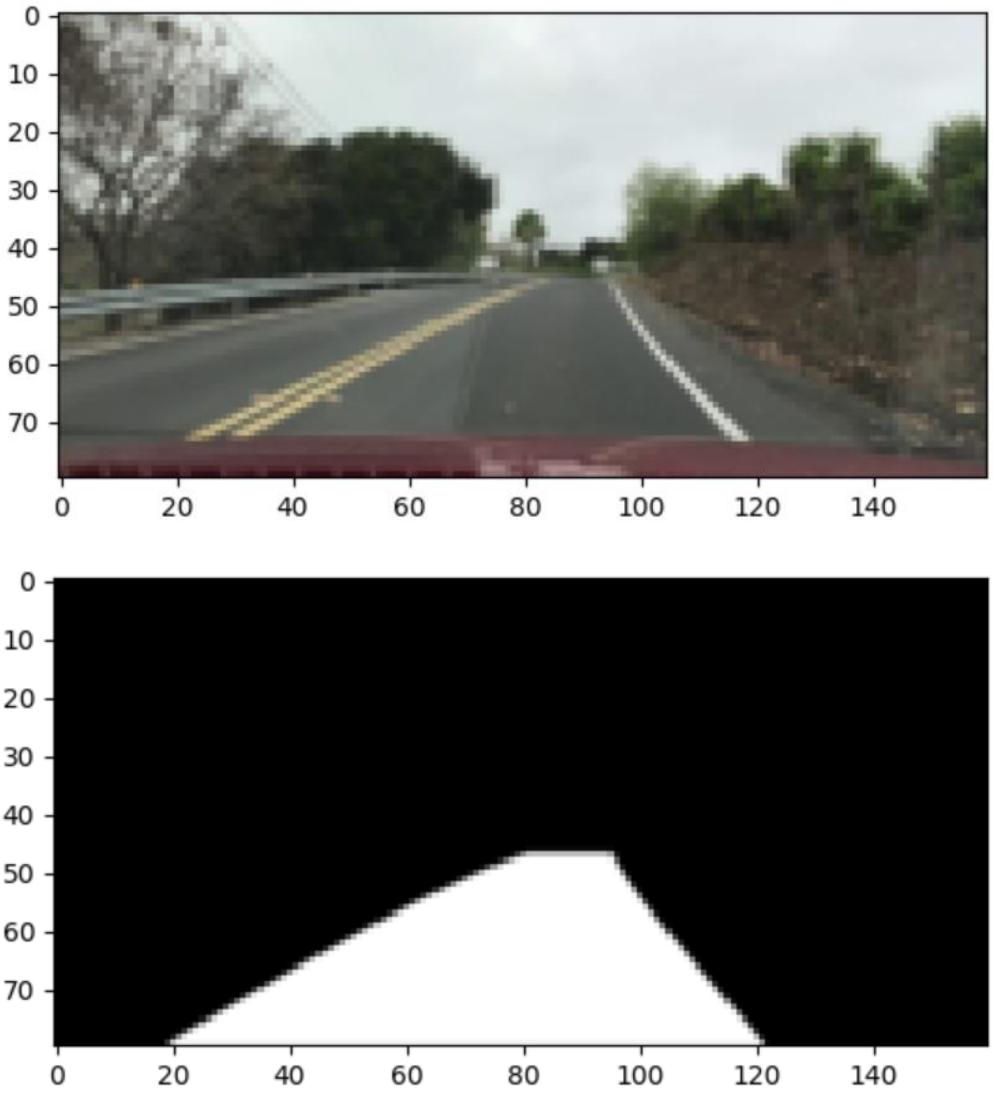


Рис 2. Пример набора данных

Но для удобства обработки данных набор данных сохраняется и импортируется в формате файла pickle.

2. Построить модель нейронной сети

Установить необходимые зависимости и импортировать необходимые библиотеки.

```
!pip install -q moviepy
!apt install imagemagick
!pip install imageio==2.4.1
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
imagemagick is already the newest version (8:6.9.7.4+dfsg-16ubuntu6.14).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: imageio==2.4.1 in /usr/local/lib/python3.8/dist-packages (2.4.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from imageio==2.4.1) (1.21.6)
Requirement already satisfied: pillow in /usr/local/lib/python3.8/dist-packages (from imageio==2.4.1) (7.1.2)
```

```
import numpy as np
import pickle
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Activation, Dropout, UpSampling2D, concatenate
from keras.layers import Conv2DTranspose, Conv2D, MaxPooling2D
from tensorflow.keras.layers import BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from keras import regularizers
import numpy as np
import cv2
from scipy import misc
from skimage.transform import resize
from IPython.display import HTML
from keras.models import load_model
import matplotlib.pyplot as plt
from moviepy.editor import VideoFileClip
```

Импортировать набор данных

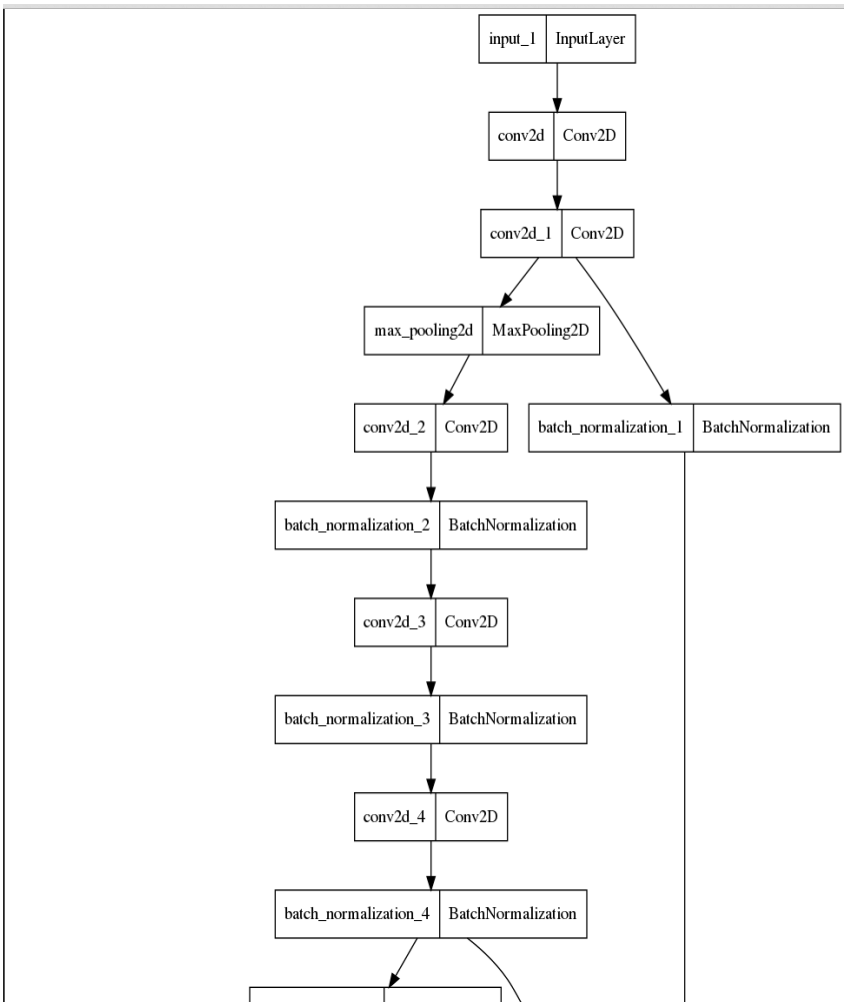
```
!wget https://www.dropbox.com/s/rrh8lrdclzlnxzv/full_CNN_train.p?dl=0
!wget https://www.dropbox.com/s/ak850zqqfy6ily0/full_CNN_labels.p?dl=0
```

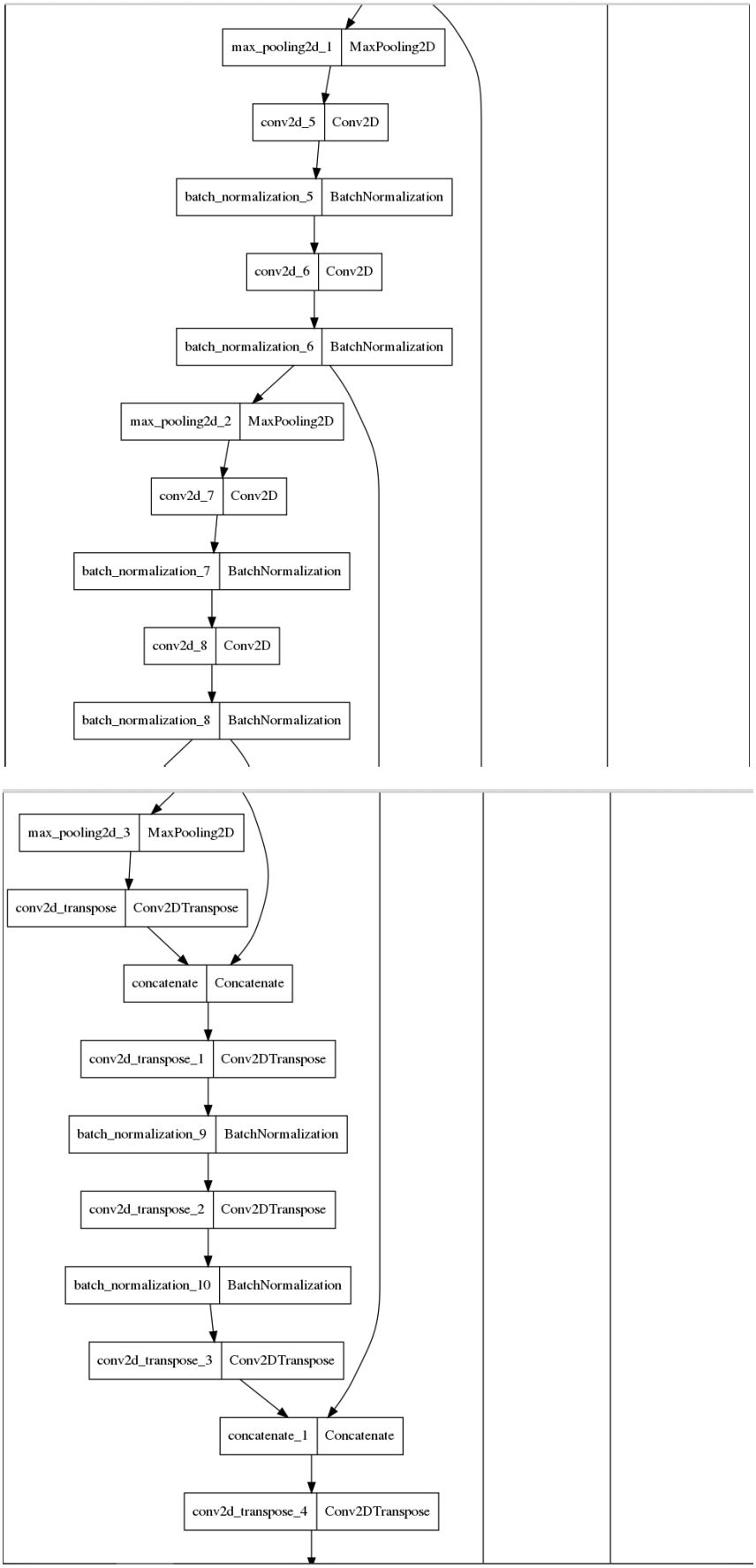
```
--2022-12-26 14:29:00-- https://www.dropbox.com/s/rrh8lrdclzlnxzv/full_CNN_train.p?dl=0
Resolving www.dropbox.com (www.dropbox.com)... 162.125.81.18, 2620:100:6031:18::a27d:5112
Connecting to www.dropbox.com (www.dropbox.com)|162.125.81.18|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: /s/raw/rrh8lrdclzlnxzv/full_CNN_train.p [following]
--2022-12-26 14:29:01-- https://www.dropbox.com/s/raw/rrh8lrdclzlnxzv/full_CNN_train.p
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
```

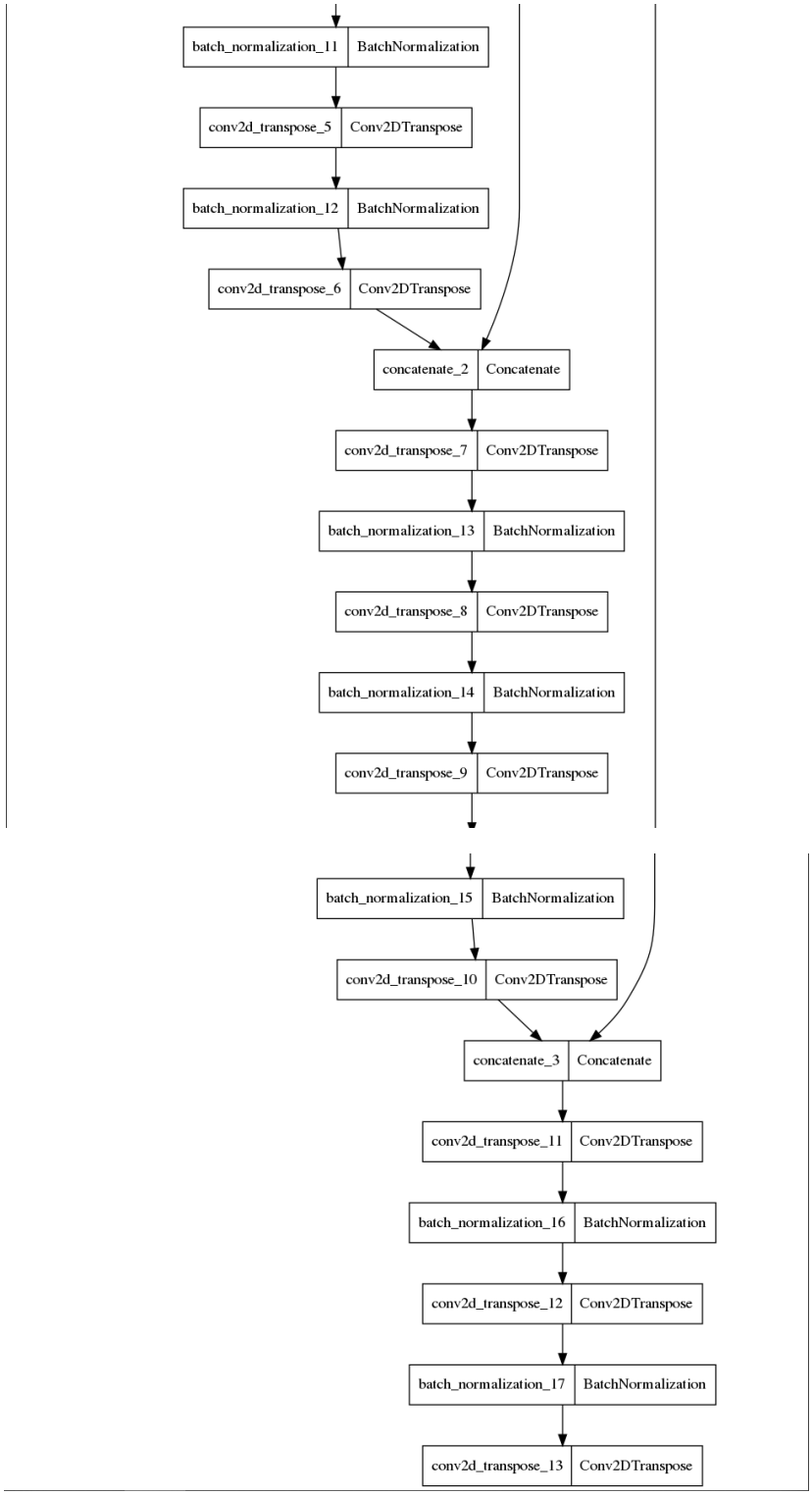
```
# Load training images
train_images = pickle.load(open("full_CNN_train.p", "rb" ))

# Load image labels
labels = pickle.load(open("full_CNN_labels.p", "rb" ))

train_images = np.array(train_images)
labels = np.array(labels)
```

При определении структуры сети увеличивается количество слоев сети и изменяется количество ядер свертки каждого слоя. Вдохновленная сетью Unet, структура параллельного перехода используется для соединения карты признаков кодирования и карты признаков декодирования, чтобы можно было использовать несколько уровней информации для предсказания классификации. Измените UpSampling2D на Conv2DTranspose, чтобы реализовать процесс повышения дискретизации. UpSampling2D напрямую использует исходное значение пикселя, чтобы заполнить процесс без обучения, в то время как Conv2DTranspose имеет процесс обучения, и эффект лучше.

Структура сети показана на рисунке.




max_pooling2d_1 MaxPooling2D
conv2d_5 Conv2D
batch_normalization_5 BatchNormalization
conv2d_6 Conv2D
batch_normalization_6 BatchNormalization
max_pooling2d_2 MaxPooling2D
conv2d_7 Conv2D
batch_normalization_7 BatchNormalization
conv2d_8 Conv2D
batch_normalization_8 BatchNormalization
max_pooling2d_3 MaxPooling2D
conv2d_transpose Conv2DTranspose
concatenate Concatenate
conv2d_transpose_1 Conv2DTranspose
batch_normalization_9 BatchNormalization
conv2d_transpose_2 Conv2DTranspose
batch_normalization_10 BatchNormalization
conv2d_transpose_3 Conv2DTranspose
concatenate_1 Concatenate
conv2d_transpose_4 Conv2DTranspose



Результаты обучения модели следующие:

```
Epoch 1/15
<ipython-input-3-b3cc1317ede1>:159: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future ver
  model.fit_generator(datagen.flow(X_train, y_train, batch_size=batch_size), steps_per_epoch=len(X_train)/batch_size,
717/717 [==============================] - 40s 38ms/step - loss: 0.0199 - val_loss: 0.0081
Epoch 2/15
717/717 [==============================] - 23s 32ms/step - loss: 0.0050 - val_loss: 0.0045
Epoch 3/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0040 - val_loss: 0.0038
Epoch 4/15
717/717 [==============================] - 23s 31ms/step - loss: 0.0036 - val_loss: 0.0042
Epoch 5/15
717/717 [==============================] - 22s 31ms/step - loss: 0.0033 - val_loss: 0.0029
Epoch 6/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0029 - val_loss: 0.0028
Epoch 7/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0024 - val_loss: 0.0025
Epoch 8/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0023 - val_loss: 0.0035
Epoch 9/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0025 - val_loss: 0.0024
Epoch 10/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0020 - val_loss: 0.0020
Epoch 11/15
717/717 [==============================] - 22s 31ms/step - loss: 0.0018 - val_loss: 0.0028
Epoch 12/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0017 - val_loss: 0.0022
```


```
Epoch 12/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0017 - val_loss: 0.0022
Epoch 13/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0016 - val_loss: 0.0020
Epoch 14/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0015 - val_loss: 0.0019
Epoch 15/15
717/717 [==============================] - 22s 30ms/step - loss: 0.0014 - val_loss: 0.0018
Model: "model"
```


Определить функцию входного тестового видео, и результат обнаружения каждого изображения кадра будет средним из результатов обнаружения предыдущих 5 изображений кадра.

```
model = load_model('full_CNN_model_HYe15.h5')
class Lanes():
        def __init__(self):
                self.recent_fit = []
                self.avg_fit = []

def road_lines(image):
        """ Takes in a road image, re-sizes for the model,
        predicts the lane to be drawn from the model in G color,
        recreates an RGB image of a lane and merges with the
        original road image.
        """

        # Get image ready for feeding into model
        small_img = resize(image, (80, 160, 3))
        small_img = np.array(small_img)
        small_img = small_img[None,:,:,:]

        # Make prediction with neural network (un-normalize value by multiplying by 255)
        prediction = model.predict(small_img)[0] * 255
        # Add lane prediction to list for averaging
        lanes.recent_fit.append(prediction)

        # Only using last five for average
        if len(lanes.recent_fit) > 5:
                lanes.recent_fit = lanes.recent_fit[1:]
```

```
        # Calculate average detection
        lanes.avg_fit = np.mean(np.array([i for i in lanes.recent_fit]), axis = 0)

        # Generate fake R & B color dimensions, stack with G
        blanks = np.zeros_like(lanes.avg_fit).astype(np.uint8)

        lane_drawn = np.dstack((blanks, lanes.avg_fit, blanks))

        # Re-size to match the original image
        lane_image = resize(lane_drawn, (720, 1280, 3))

        #plt.imshow(lane_image)
        #plt.show()


        # Merge the lane drawing onto the original image
        result = cv2.addWeighted(image, 1, lane_image, 1, 0, dtype=cv2.CV_8UC3)

        return result

lanes = Lanes()

# Where to save the output video
vid_output = '/content/gdrive/MyDrive/test_video_hy_1.mp4'

# Location of the input video
```

```
# Location of the input video
clip1 = VideoFileClip("/content/gdrive/MyDrive/challenge.mp4")

vid_clip = clip1.fl_image(road_lines)
vid_clip.write_videofile(vid_output, audio=False)
```

```
78%|████████  | 197/251 [01:16<00:21,  2.53it/s]1/1 [==============================] - 0s 26ms/step
79%|████████  | 198/251 [01:17<00:20,  2.58it/s]1/1 [==============================] - 0s 28ms/step
79%|████████  | 199/251 [01:17<00:20,  2.51it/s]1/1 [==============================] - 0s 24ms/step
80%|████████  | 200/251 [01:18<00:20,  2.51it/s]1/1 [==============================] - 0s 22ms/step
80%|████████  | 201/251 [01:18<00:19,  2.57it/s]1/1 [==============================] - 0s 23ms/step
80%|████████  | 202/251 [01:18<00:18,  2.59it/s]1/1 [==============================] - 0s 27ms/step
81%|████████  | 203/251 [01:19<00:19,  2.52it/s]1/1 [==============================] - 0s 25ms/step
81%|████████  | 204/251 [01:19<00:18,  2.53it/s]1/1 [==============================] - 0s 25ms/step
82%|████████  | 205/251 [01:19<00:17,  2.60it/s]1/1 [==============================] - 0s 31ms/step
82%|████████  | 206/251 [01:20<00:17,  2.58it/s]1/1 [==============================] - 0s 41ms/step
82%|████████  | 207/251 [01:20<00:17,  2.52it/s]1/1 [==============================] - 0s 22ms/step
83%|████████  | 208/251 [01:21<00:16,  2.56it/s]1/1 [==============================] - 0s 21ms/step
```

Результаты теста следующие:

3. Заключение

Эффект обнаружения полос с помощью глубокого обучения хороший, и он отличается от традиционного метода cv, который требует ручного разделения области, что больше подходит для обнаружения в реальном времени. В то же время, изменив механизм внимания и определив больше областей, можно одновременно распознавать транспортные средства, полосы движения и пешеходов.

Список использованных источников

[1] Badrinarayanan V., Kendall A., Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation //IEEE transactions on pattern analysis and machine intelligence. – 2017. – Т. 39. – №. 12. – С. 2481-2495.

[2] Tang J., Li S., Liu P. A review of lane detection methods based on deep learning //Pattern Recognition. – 2021. – Т. 111. – С. 107623. [3] Bochkovskiy A., Wang C. Y., Liao H. Y. M. Yolov4: Optimal speed and accuracy of object detection //arXiv preprint arXiv:2004.10934. – 2020.

[3] Assidiq A. A. M. et al. Real time lane detection for autonomous vehicles //2008 International Conference on Computer and Communication Engineering. – IEEE, 2008. – С. 82-88.