과제3 코랩링크

https://colab.research.google.com/drive/1dnN-25Rj3rhms4B5nsyCs31X8C7X5c9O?usp=sharing

구현코드

```
def inference(gt_img_org):
    # BGR 순서
    org_shape = gt_img_org.shape
    gt_image = cv2.resize(gt_img_org, dsize=(512, 256), interpolation=cv2.INTER_LINEAR)
    gt_image = gt_image / 127.5 - 1.0
    gt_image = torch.tensor(gt_image, dtype=torch.float)
    gt_image = np.transpose(gt_image, (2, 0, 1))
    gt_image = gt_image.to(device)
    # lane segmentation
    binary_final_logits, instance_embedding = LaneNet_model(gt_image.unsqueeze(0))
    binary_final_logits, instance_embedding = binary_final_logits.to('cpu'), instance_embedding.to('cpu')
    binary_img = torch.argmax(binary_final_logits, dim=1).squeeze().numpy()
    binary_img[0:65,:] = 0

    # lane clustering & segemented frame embedding
    rbg_emb, cluster_result = process_instance_embedding(instance_embedding, binary_img,
                                                          distance=1.5, lane_num=4)

    rbg_emb = cv2.resize(rbg_emb, dsize=(org_shape[1], org_shape[0]), interpolation=cv2.INTER_LINEAR)
    a = 0.6
    frame = a * gt_img_org[..., ::-1] / 255 + rbg_emb * (1 - a)
    frame = np.rint(frame * 255)
    frame = frame.astype(np.uint8)

    return frame
```

```
def video2segemented_video(video_path):

    # TODO: video to frames
    vidcap = cv2.VideoCapture(video_path)
    success,image = vidcap.read()
    frames = []
    while (success):
      success,image = vidcap.read()
      if not success : break

    # TODO: extract lane from frame
      img = inference(image)
      frames.append(img)

    # TODO: frames to video & store video
    fps = 30
    height, width, layers = frames[0].shape
    size = (width, height)
    out = cv2.VideoWriter(video_path,cv2.VideoWriter_fourcc(*'DIVX'), fps, size)
    for i in range(len(frames)) :
      out.write(frames[i])
    out.release()
    pass

# Test
video_path =  "/content/drive/MyDrive/lanenet/lane/운전영상.mp4"

video2segemented_video(video_path)
```

구현코드 설명

#video_cap으로 받은 동영상 경로를 받아서 프레임으로 분할을 해 줍니다.

```
    vidcap = cv2.VideoCapture(video_path)
    success,image = vidcap.read()

    frames = []

    while (success):

      success,image = vidcap.read()

      if not success : break
```

#분할한 프레임을 차선 검출 알고리즘으로 구현된 inference에 넣어 프레임에 대해 차선을 검출해 줍니다. 그리고 frames에 차선이 검출된 프레임을 넣어줍니다.

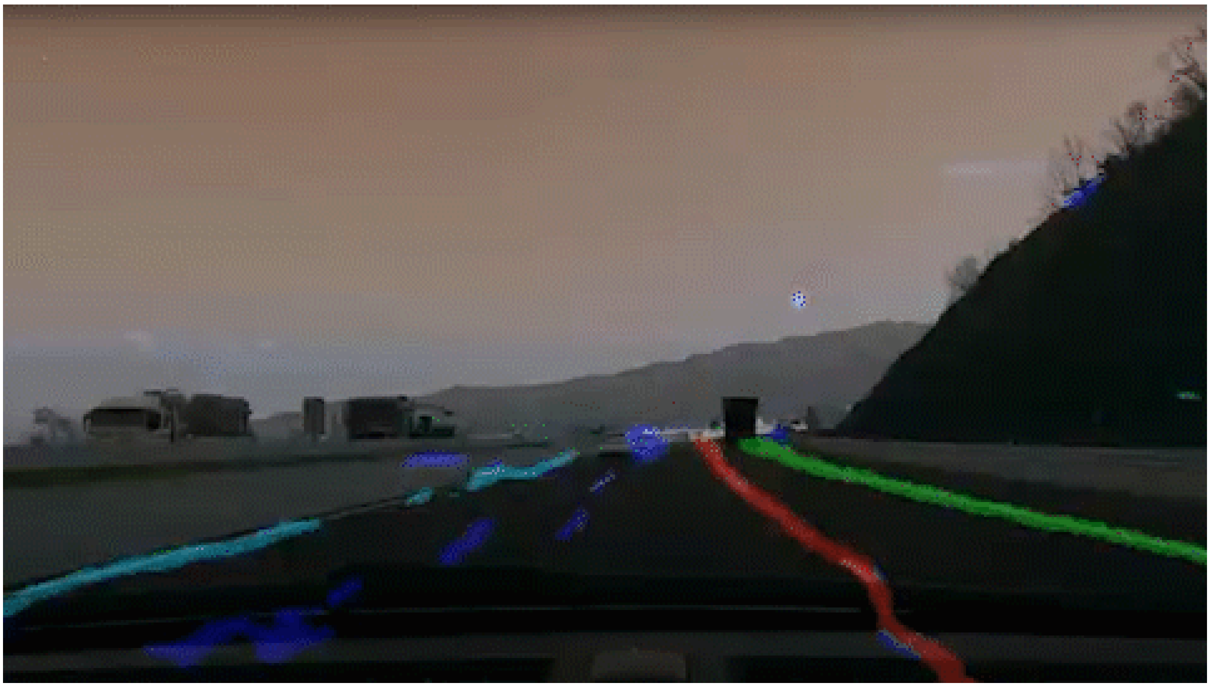
```
      img = inference(image)

      frames.append(img)
```

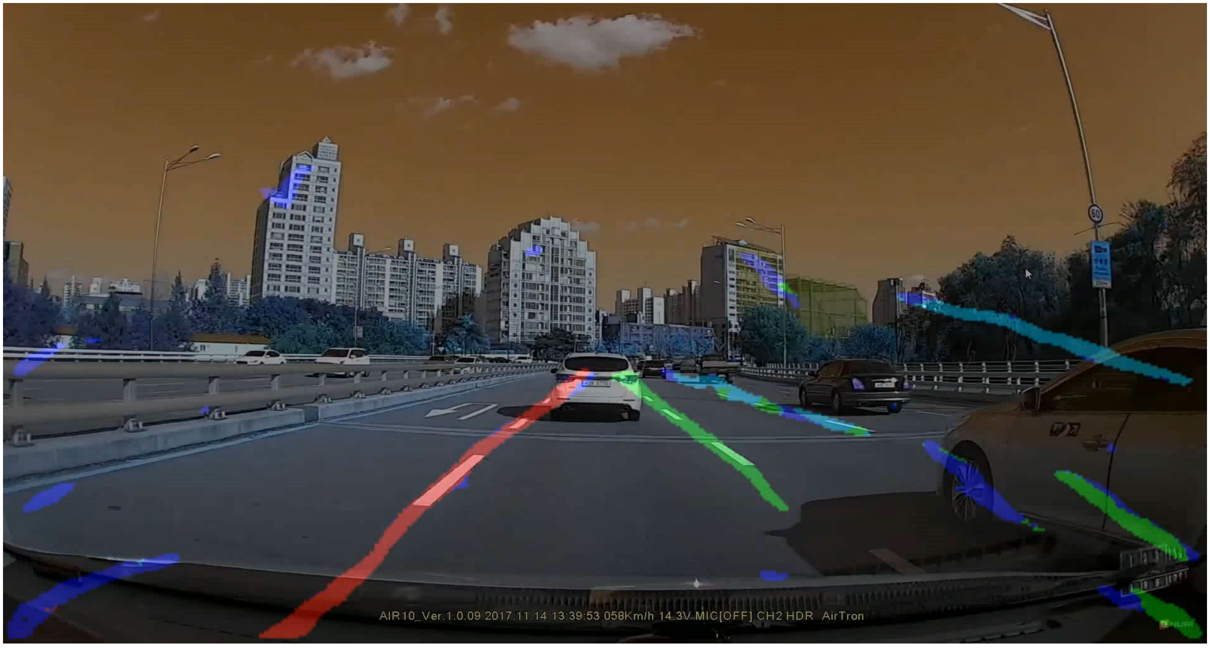
#차선이 검출된 프레임을 동영상으로 만들어 줍니다.

```
      fps = 30

      height, width, layers = frames[0].shape

      size = (width, height)

      out = cv2.VideoWriter(video_path,cv2.VideoWriter_fourcc(*'DIVX'), fps, size)

      for i in range(len(frames)) :

        out.write(frames[i])

      out.release()

    pass
```

실행결과(영상주소)

https://drive.google.com/file/d/1LPI-RqQ2LjBedO-WbijudOhNtZUYYuuy/view?usp=sharing




AIR10_Ver.1.0.09 2017.11.14 13:39:53 058Km/h 14.3V MIC[OFF] CH2 HDR AirTron