안녕하세요 저희는 뉴럴 스타일 트랜스퍼라는 주제를 선정한 13조 안의진 박찬영이고 저는 발표는 맞게 된 안의진 입니다. 저희는 주제 선정에 있어서 사람 들에게 비쥬얼하고 자신만의 작품을 만들어 볼 수 있는 툴을 소개해주고 싶어 뉴럴 스타일 트랜스퍼라는 주제를 선정하게 되었습니다. Style transfer란 일반사진인 content image를 유명한 화가의 작품인 Style image와 결합해 일반사진을 마치 유명화가가 그린 것처럼 변화해주는 머신러닝 프로젝트입니다. 이처럼 이 기술을 이용하면 저희가 찍은 일반 사진들도 딥러닝 학습을 통해 화가의 그림처럼 transfer 할 수 있습니다. 하지만 이러한 작업이 단순한 작업은 아닙니다. 이러한 결합은 하기 위해선 CNN을 이용해야 합니다.

다음 그림은 뉴럴 스타일 트랜스퍼에 사용되는 컨벌루션 뉴럴 네트워크 입니다. 레이어 1에서는 32개의 필터를 사용하면서 단순한 패턴을 캡처합니다. 예를 들어 수직선이나 직선처럼 단순하지만 중요한 패턴을 캡처합니다. 그 다음에는 64의 필터를 가지고 있는 레이어2로 넘어갑니다. 이 레이어부터는 개의 얼굴이나 자동차의 바퀴 같은 좀더 복잡한 특성들도 캡처합니다. 이러한 단순한 패턴과 복잡한 패턴을 다르게 캡처하는 것을 feature representation이라고 합니다. 여기서 중요한 것은 CNN 그 자체로는 이미지가 무엇인지 알지 못하지만 특정 이미지가 나타내는 중요한 것을 인코딩하는 법을 배웁니다. 이제 다음으로는 CNN이 어떻게 컨텐츠와 스타일을 이미지에서 캡처해 내는지를 살펴볼 것입니다.

이것은 VGG199라는 뉴얼 스타일 트랜스퍼에서 사용하는 네트워크 입니다. 이것은 통해 CNN은 수많은 이미지들은 이미지넷 데이터 베이스에서 가져와 train 시킵니다. 이 네트워크는 이름에서 알 수 있듯이 19의 레이어를 통해 수만개의 이미지를 train시켜 이미지의 좀더 고급지고 복잡한 특성까지 캡처할 수 있게 합니다.

다음으로 이제 cnn의 key 파트인 인코딩 부분에 대해 살펴볼 것입니다. 먼저 우리는 노이즈 이미지를 initialize 합니다. 이 이미지는 우리의 아웃풋 이미지가 될 것입니다. 그런 다음 이 이미지가 네트워크의 특정 계층의 컨텐츠 및 스타일 이미지와 얼마나 유사한지 계산합니다. 우리는 아웃풋 이미지에 content image와 style image의 합작품이 있어야 하기 때문에 content와 style 두 이지미의 loss 즉 손실을 계산합니다. 이러한 과정을 거쳐서 결과물인 그림이 나오게 되는 것입니다.

다음으로 content loss에 대해 살펴보겠습니다. 앞에서도 말했듯이 content loss을 계산한다는 것은 랜덤하게 만들어진 노이즈 이미지가 content이미지와 얼마나 유사한지를 계산하는 것을 말합니다. 손실을 계산하기 위해 pre-trained된 네트워크에서 숨겨진 계층 (L)을 선택한다고 가정합니다. 그다음 P와 F를 원본 이미지와 생성된 이미지로 만듭니다. 그리고 F [l]와 P [l]은 레이어 L의 각 이미지의 특성을 나타냅니다. 이러한 content loss의 계산식은 다음과 같습니다.

다음으로 Style loss를 계산한다는 것은 어떻게 우리가 이미지의 스타일을 캡처할 것인가를 말합니다. 이미지에서 특정한 layer L를 선택했다고 하면 그 layer는 여러 다른 채널들, 특성, 또한 필터들을 보여줄 것입니다. 우리가 이러한 이미지에서 스타일을 캡처 하기 위해서는 여러 특성들과 필터들이 어떠한 상관관계에 있는지 계산해야 합니다. 이러한 계산을 위하여 우리는 Gram Matrix를 사용하여 상관관계를 판별하고 이를 이용해 여기에 나와있는 style cost function 식을 구현합니다.

이렇게 content loss와 style loss를 계산한 다음에 두개의 합해주면 Total loss를 구할 수 있습니다. 식에 보이는 알파와 베타는 일반적으로 생성된 아웃풋 이미지에 cost에 대한 가중치를 나타냅니다. 이런 식으로 손실이 계산되고 나면 우리가 배운 역전파를 이용해 손실을 최소화하고 역전파를 이용하고 나면 랜덤하게 만들어진 이미지가 아름다운 명화가 되어있는 모습을 보실 수 있을 겁니다.

이것으로 style transfer에 대한 소개는 마치고 이제는 한번 실제로 구현을 통해 style transfer가 이루어지는 과정을 살펴보도록 하겠습니다.

제가 이 작업을 jupiter노트북과 colab 두 군대에서 실행시켜 보았는데 Jupiter 노트북의 환경보다 colab이 좀더 수월하게 작동하는 것을 볼 수 있었고 이점을 참조해주시면 좋겠습니다.

먼저 인터넷 상에서 이미지를 다운로드 합니다. 저는 거북이와 일본화가의 그림을 다운받도록 하겠습니다. 그 다음 이미지를 학습시킬 때 필요한 모듈들을 import해줍니다. 그다음 load img와 imshow를 통하여 다운 받은 이미지를 볼수 있게 10,10 사이즈로 비주얼 라이즈 시켜줍니다. 보시면 거북이 이미지와 유명화가의 그림이 보여지는 것을 볼 수 있습니다.

다음으로 이미지를 처리하기 편한 형식으로 바꾸어 줍니다. 이 방식은 vgg training 과정과 동일한 과정입니다. 그리고 우리는 이미지가 최적화가 잘 되어있는 확인 하는 방법으로 위에서 해주었던 과정을 반대로 해줍니다.

다음으로 우리는 content와 style의 필요없는 중간 계층을 제거해주고 필요한 부분만 남깁니다.

이제 앞에서 설명했듯이 vgg19를 이용해 content, style, 노이지 이미지에 특성을 추출해낸다. 다음으로 아웃풋 이미지를 만들기 위해 필요한 content loss와 style loss를 계산해줍니다. 보시면 앞서 설명 드린 것 같이 그램 매트릭스를 이용하여 style loss를 계산하는 것을 볼 수 있습니다. 다음으로 경사 하강법/역전파를 이용해 loss를 최소화 시키는 작업을 진행한다음 총 loss와 경사를 계산해 줍니다. 그런 다음 마지막으로 최적화시켜주는 작업을 진해 해주면 거북이 그림과 일본화가의 그림이 합쳐진 결과물이 나타나게 됩니다.

밑에 보시면 제가 불러온 거북이 이미지 말고도 어떠한 이미지와 도 잘 작동하는 것을 볼 수 있습니다. 여러분들도 남은 학기 잘 마무리하시고 방학 때 시간이 나면 한번 여러분만의 이미지와 유명 화가의 그림을 합쳐서 확인해 보시는 것도 좋을 것 같습니다. 이상으로 저희 13조의 발표를 마치겠습니다 들어 주셔서 감사합니다.