Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations

Abstract

- 비디오에서 높은 퀄리티의 face editing은 많은 문제를 몰고 온다.
- 본 논문에서는 현재의 facial editing method와 여러 characteristic artifacts를 리뷰하고, 상대적으로 단순한 시각적 아티펙트가 그러한 조작임을 노출하는 데 효과적임을 확인하였다.
- 이러한 method가 visual features에 기반하였기에, 굉장히 간단하다. 하지만 간단함에도 불구하고 좋은 성능을 보인다.

Introduction

근래에 딥페이크 이미지 및 비디오 등이 미디어에 자주 노출되며 우리는 “과연 저 영상이나 이미지를 신뢰할 수 있는가”에 대한 질문을 하게 되었다. 그리고 딥페이크 기술이 발전함에 따라 진짜와 구분하기 어려운 딥페이크 이미지 및 비디오가 생성된다.

본 논문에서는 이미지 포렌식(과학수사로도 볼 수 있음)과 관련된 연구를 리뷰하고(본 논문 리뷰에서는 정리 X), 얼굴 이미지의 자동 생성 및 편집 방법을 제시한다. 그 후, 위의 방법에서 나올 수 있는 결함들을 조사하였다.

Manipulation Artifacts

딥페이크 기술들이 발전함에 따라 진짜 이미지와 가짜 이미지를 구분하기가 어려워 졌지만, 몇가지의 visual artifact들이 존재한다.

Global Consistency

이미지 생성 method(GAN 등)는 생성 모델의 잠재 공간과 특정 이미지 간의 보간을 부드럽게 하는 곳에 쓰일 수 있다. 또한 랜덤한 얼굴을 생성하는데 사용될 수 있다. 두 얼굴 간의 보간을 진행할 때에는 대부분 데이터가 의미있게 생성된다. 하지만 새로운 얼굴을 생성하는 작

업에서는 이이지 보간에 사용되는 데이터 포인트는 랜덤하고 꼭 의미있는 데이터 포인트라고 할 수 없다.

Illumination Estimation

여러 attribute로 얼굴을 re-rendering 할 때에, 입사 조명은 원본 이미지에서 위조 이미지로 transfered 되어야 한다.

Diffuse reflection은 보통 잘 reproduce 된다. 특별히 딥러닝 기술을 통한 manipulation에서 본 논문의 저자들을 연관된 artifact를 찾아내기가 어려웠다고 한다.

다음으로 Face2Face manipulation의 몇 몇 케이스에서 shading artifact들이 나타날 수 있다. 그 artifact는 보통 코 부분에서 발생한다(사이드 부분의 render이 너무 어두운 문제).

본 논문의 저자들을 이러한 현상은 제한된 Face2Face illumination model 때문이라고 가설을 세웠다.

얼굴의 정반사는 눈 부분에서 가장 눈에 띈다. 딥페이크 기술을 사용해 생성된 이미지는 정반사에 대해 unconvincing하다. 눈 부분의 반사는 없거나, 하얀 얼룩으로 나타난다. 이 결함은 전체적인 가짜 눈의 모습이 둔한 것처럼 보이게 한다.

Geometry Estimation

얼굴 이미지를 조작하려면 안면 형상을 추정해야 한다. 이전의 illumination의 경우에는, Face2Face는 변형 가능한 모델을 이미지에 피팅시켜 형상 추정을 모델링한다. 딥러닝 기반의 기술은 데이터에서 기본적인 모델을 학습 가능하다.

Face2Fac2 데이터에서 기본 형상에 대한 부정확한 추정으로 발현되는 artifact를 찾을 수 있었다. 얼굴 위에 mask 값을 줄 때에 만약 추정에 오류가 있다면 문제가 나타난다는 것이다. 특히 이 문제는 코 부분, 얼굴 경계, 눈썹에서 많이 나타난다. 소셜 미디어에서도 이처럼 치아 등에서 결함이 발생함을 볼 수 있을 것이다.

Classification Based on Visual Artifacts

딥페이크 이미지에서 artifact가 언제나 구분 가능하도록 잘 보이는 것은 아니다. 아무튼 간단한 특징을 찾는 것으로도 딥페이크 이미지를 구분하는 것이 가능할 수 있다.

Detection Pipelines

Generated Faces

여러 눈의 색 정보는 딥페이크 이미지를 판단하는 것에 사용된다. 각 눈에서 색 정보를 추출하기 위해 본 논문의 저자들은 cv method를 사용했다. 전체 이미지에서 얼굴 부분을 잘라내고 resize 했다. Segmentation에서는 홍채 부분을 잘라내었다. 마지막으로 Canny edge detection과 Hough Circle Transformation이 적용되었다.

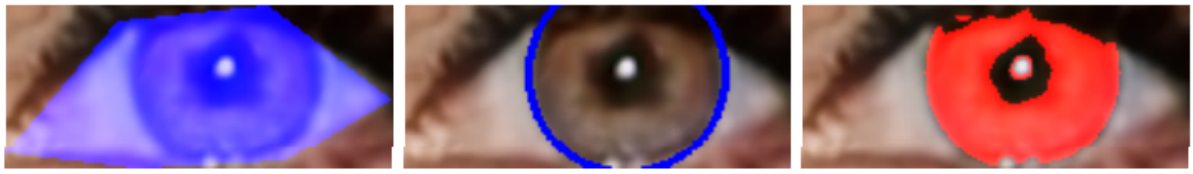

Figure 9. Example result of the iris segmentation.

홍채 부분에서 중심 부분과의 거리를 계산해 양 눈을 비교함으로 이미지를 분류한다.

본 논문에서는 양 눈의 dissimilarity를 특성화하는 여러 feature을 정의하였다. 먼저, 색 공간을 HSV로 바꾸어 segment 된 양 눈의 pixel을 average 했다.

$$\begin{aligned}
\mathrm{Dist}_H &= \min(|l_H - r_H|, 360 - |l_H - r_H|) \\
\mathrm{Dist}_S &= |l_S - r_S| \\
\mathrm{Dist}_V &= |l_V - r_V| \\
\mathrm{Dist}_{HSV} &= \mathrm{Dist}_H + \mathrm{Dist}_S + \mathrm{Dist}_V \quad .
\end{aligned}$$

- l_x, r_x : 왼쪽, 오른쪽

이를 통해 본 논문의 저자들은 6차원의 feature을 만들어냈다.

$$F = (\mathrm{Dist}_H, \mathrm{Dist}_S, \mathrm{Dist}_V, \mathrm{Correl}_R, \mathrm{Correl}_G, \mathrm{Correl}_B)$$

이 feature은 knn 모델로 보내져 계산된다.

Deepfakes

딥페이크 이미지를 구분 할 때에 눈과 이의 디테일을 볼 수 있다. 본 논문의 저자들은 얼굴의 랜드마크를 찾아 crop하여 사용하였다. 여러 resolution의 이미지가 있기에 이미지들을 resize 해서 사용하였다. 구 중 이 부분을 segment 하기 위해 이미지를 grayscale로 바꾸어 주었다. 그 후 K-Means clustering을 사용해 밝은 부분과 어두운 부분으로 구분하였다. 그 중 밝은 부분은 이 부분으로 판단 하였다. 샘플 중 너무 적은 부분이 이로 판단되면 그 샘플은 무시하였다.

또한 본 논문의 저자들은 texture energy approach를 사용해 texture의 complexity를 설명하는 feature을 뽑아내었다.

각 샘플에서 feature vector을 뽑아내기 위해 눈, 이, 전체 이미지에서 뽑아낸 9개의 feature을 평균한다. 그리고 2개의 모델을 통해 구분을 진행한다. 첫 모델은 logistic regression이고, 두 번째 모델은 간단한 neural network가 사용되었다.

Face2Face

Face2Face 데이터에서도 위의 모델과 같은 모델을 사용하였다(feature은 다름). 이 데이터셋에서는 눈과 이가 아닌 face border과 nose tip을 확인하였다.

개선 방안

- 눈, 코, 입 외에 다른 피부 등에서의 특징을 찾아낼 수 있다면 더욱 좋은 성능의 딥페이크 이미지 검출 모델을 완성할 수 있을 것같다.