Masked face detection Using CNN & Machine Learning

Yejin Hwang
Ewha Womans University
yejinhwang@ewhain.net

Abstract

코로나 19 바이러스의 유행이 2 년 이상 지속되면서 일상생활에서 마스크 착용은 필수적인 일이 되었다. 최근 사회적 거리두기와 실외 마스크 필수 착용이 해제되는 등 방역조치의 완화가 이루어지고 있으나, 여전히 실내에서는 필수적으로 마스크를 착용해야 한다. 따라서 본 연구에서는 최근의 CNN 방법론과 boosting 및 stacking을 결합한 모델을 통해 사람의 얼굴 이미지가 주어졌을 때 그 사람의 마스크 착용 여부를 예측하였다. CNN을 통해 feature extraction을 우선적으로 진행하였으며 DenseNet, MobileNetV2, Efficient-NetV2 를 각각 사용하여 결과를 비교하였다. 이후 classification 과정에서는 CNN 모델로 마스크 착용 여부에 대한 예측까지 end-to-end로 진행하여 얻은 예측치와, feature에 XGBoost, LightGBM을 적용하여 각각 얻은 예측치를 stacking하여 최종적인 예측치를 계산하였다. Test accuracy 및 confusion matrix를 통해 성능을 확인하였을 때 DenseNet으로 feature extraction 과정을 거친 뒤 3 가지 classification에 대한 예측치를 stacking한 방법의 test accuracy가 0.993 으로 높은 정확도를 얻을 수 있었다.

1. Introduction

2020 년 우리나라에서 코로나 19 첫 확진자 발생 이후 집단 감염이 계속해서 발생하고 델타 변이, 오미크론 변이와 같은 여러 변이 바이러스가 등장하는 등 현재까지도 바이러스의 유행이 계속되고 있다. 이러한 상황이 지속되면서 코로나 19 의 주 감염 경로인 비말을 통한 전파를 막아주는 역할을 하는 마스크 착용은 많은 사람들에게 일상적인 일로 자리잡았다.

2022 년 4 월 18 일 이후로 사회적 거리두기 조치가 해제되고, 5 월 2 일 이후 실외 마스크 필수 착용이 해제되는 등 최근 방역조치의 완화가 이루어지고 있다. 그러나 여전히 하루에 수천명의 확진자가 발생하고 있어 방역 당국에서는 실내 마스크 의무 착용은 유지되어야 한다는 입장을 보이고 있다.

이러한 상황에서 실내 시설을 이용하는 사람들에 대해 사람들이 마스크를 잘 착용하였는지 확인하는 것은 코로나 19 바이러스의 전파를 예방하는 역학을 할 수 있을 것이다. 따라서 본 연구에서는 사람의 얼굴 이미지 데이터가 주어졌을 때 그 사람이 마스크 착용 여부에 대해 예측하는 모델을 만들어보고자 하였다.

2. Related work

딥러닝에서 face recognition과 관련해서는 비교적 이전부터 연구가 진행되어 왔으나, masked face detection과 관해서는 코로나 19 의 유행 이후에 여러 방법론들이 제안되고 있어 어떤 하나의 논문이 SOTA라고 칭해지기 보다는 다양한 구조를 가진 모형을 제안하는 논문들이 나오고 있다.

2.1. Face recognition

2015 년에 나온 [1]은 유클리드 공간으로 이미지를 mapping하여 이미지 간의 거리를 기반으로 face similarity를 계산하고 클래스를 분류하는 FaceNet 모델을 제안하였다. Feature 학습 과정에서 triplet

loss를 사용하여 특정 사람과 같은 사람(positive)과의 거리는 가깝게, 다른 사람(negative)과의 거리는 멀어지도록 하였다.

2018 년에 제안된 [2]는 softmax 함수를 대체할 수 있는 새로운 loss function을 사용한 ArcFace 모델을 제안하였다. Loss function의 계산 과정에서 Euclidean 방식의 loss를 angular 기반의 loss를 변경하여, 서로 다른 클래스 사이에 각도의 차이, 즉 margin을 주어 더 큰 격차가 발생하도록 하였다.

2.2. Masked face detection

Masked face detection과 관련하여 리뷰한 세 논문은 모두 코로나 19 유행 이후 제안된 방법론들이다. 2021 년도에 제안된 [3]에서는 마스크 착용 여부에 대한 detection을 위해 one-stage와 two-stage detection의 앙상블 모델을 제안하였다. RestNet50 을 이용한 transfer learning과 localization performance를 높이기 위한 bounding box transformation이 모델 구조에 사용되었다.

2021 년 논문인 [4]는 마스크 착용 여부에 대한 예측을 위해 ArcFace 방법론을 바탕으로 한 Multi-Task ArcFace(MTArcFace)를 제안하였다. ResNet50 을 backbone 모델로 사용하고, [2]에서 제안된 ArcFace loss와 mask probability loss를 결합한 값을 loss function으로 활용하였다.

2020 년에 나온 [5]는 딥러닝과 머신러닝의 하이브리드 모델을 제안하였다. 먼저 ResNet50 을 이용하여 이미지의 feature를 추출하였고, 이후 마스크 착용 여부에 대한 classification에서는 decision tree, SVM, ensemble 모델을 통해 각각 예측을 진행한 뒤 결과를 voting하는 stacking ensemble 방법을 사용하여 최종적으로 예측하였다.

3. Proposed approach

Section 2 에서 리뷰한 논문들을 보면 주로 pretrained CNN 모델을 이용한 transfer learning을 통해 우선적으로 이미지 데이터에서 feature extraction을 진행하였고, 이후 분류문제를 해결하기 위해 CNN이 아닌 다른 구조를 가진 모델을 함께 사용하였다.

따라서 본 연구에서는 이전 연구들의 모델 구조를 따라 CNN 모델을 통해 feature extraction 과정을 거친 뒤 여러 머신러닝 방법론을 이용하여 얻은 예측치를 이용하여 stacking ensemble을 진행하는 masked face detection model을 사용하고자 하였다. Feature extraction과 classification 과정에서 이전 연구에 사용되지 않은 3 개의 새로운 모델을 각각 이용하여 예측하였다. 3 개의 feature extraction 방법을 각각 사용해서 stacking ensemble을 통한 classification까지 거쳤을 때 어떤 CNN 모델을 backbone으로 이용하는 것이 가장 좋은 성능을 보이는지 확인하고자 하였다. 또한 stacking 하기 이전 얻어진 3 개의 classification 모델의 예측치과 stacking을 통해 얻어진 예측치의 성능이 어떤 차이가 있는지 확인하였다. 본 연구에서 사용한 Proposed model의 전체적인 구조는 Figure 1 과 같다.

3.1. Feature extraction

선행 연구의 feature extraction 과정에서 Resnet50 을 사용한 것에서 나아가 본 연구에서는 비교적 최근에 나온 CNN 모델 구조인 DenseNet, MobileNetV2, EfficientNEtV2 를 이용하여 이미지 데이터로부터 feature를 추출하였다. DenseNet[6]은 각 layer의 feature map이 모든 다른 layer의 feature map과 연결되어 있는 구조로, 적은 채널 수를 이용하여 ResNet보다 적은 파라미터 수를 가진다. MobileNetV2[7]은 채널들을 분리하여 각 채널을 개별 kernel로 convolution 한 뒤 크기가 1×1 인 convolution을 적용하여 output 채널 수를 조절하는 depthwise separable convolution의 구조를 이용한 모델로, 기존의 convolution 연산보다 연산량을 크게 감소시켰다. 또한 성능을 개선하기 위해 inverted residual block과 linear bottleneck을 사용하였다.

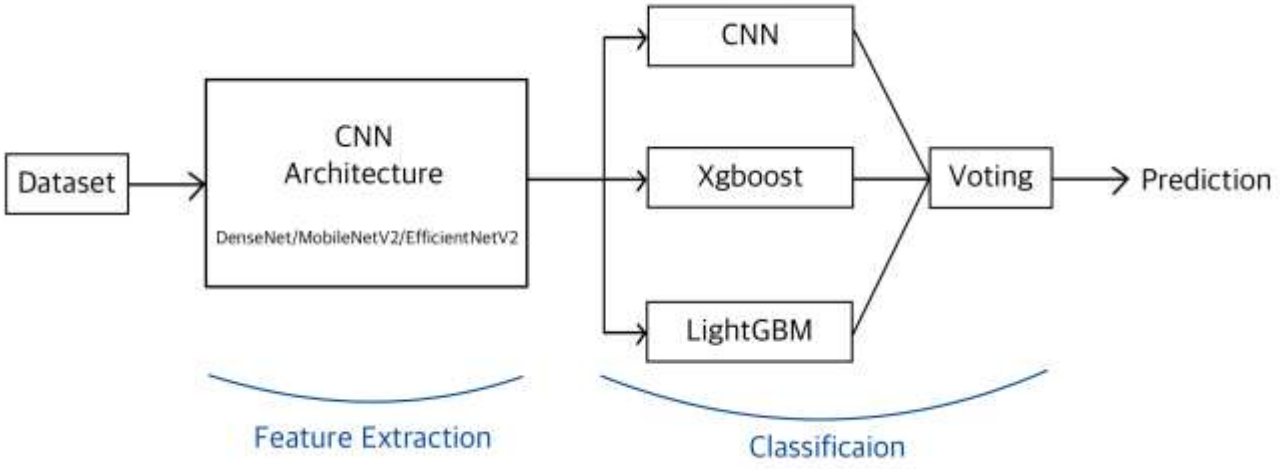

Figure 1 Proposed model architecture

EfficientNetV2[8]는 모델의 width, depth, resolution을 최적의 비율로 조합하는 compound scaling과 파라미터 수를 줄이고 학습 속도를 빠르게 하는 방법을 이용한 모델이다.

3.2. Classification

예측 단계에서도 3 가지의 방법을 사용하였다. 첫번째로는 feature extraction 단계에서 사용한 CNN 모델로 마스크 착용 여부에 대한 예측까지 진행하였다. 다른 두가지 방법은 머신러닝 방법론 중 boosting 방법을 사용하였다. Boosting은 다수의 decision tree를 적합해서 예측을 수행하는 앙상블 기법의 하나로, 약한 성능의 모델에서 시작해서 이전 단계 트리의 결과를 다음 단계 트리에 반영하여 순차적으로 예측 성능이 강한 모델을 만들어 나가는 additive model이다. 이러한 방법은 현재 tabular data를 이용한 예측에서 좋은 성능을 보이고 있으므로, boosting을 활용하여 예측을 진행하고자 하였다. 사용한 첫번째 모델은 XGBoost[9]로, 트리의 best split을 찾는 과정에서 approximate greedy algorithm을 사용하고 regularized learning objective function을 사용하는 등 효율성을 높이는 여러 기능을 포함한 gradient boosting 기반의 앙상블 방법이다. 두번째 모델인 LightGBM[9]은 XGBoost 보다도 계산과정을 줄여 학습속도를 개선한 gradient boosting 기반의 앙상블 방법이다.

4. Experiments

4.1. Implementation details

Datasets. 마스크 착용 여부를 예측하기 위해 실제 사람의 얼굴 데이터로 구성된 Real-World Masked Face Dataset[11]을 이용하여 실험을 진행하였다. 이미지 데이터는 마스크를 착용한 2203 개의 이미지와 마스크를 착용하지 않은 90468 개의 이미지로 나눠져 있다. 데이터의 균형을 맞추기 위해 90468 개의 마스크 미착용 데이터 중 착용 이미지 개수에 맞춰 2203 개의 이미지만을 random sampling하여 분석에 사용하였다. 위의 과정을 거쳐 얻은 총 4406 개의 데이터 중 전체의 80%인 3524 개의 이미지는 train set으로, 20%에 해당하는 882 개의 이미지는 test set으로 분리하였으며 분리 과정에서 label의 비율을 고려하였다. Train set을 이용하여 모델을 적합한 후 test set의 label을 예측하고, 예측한 label과 실제 label에 대한 test accuracy와 confusion matrix를 확인하여 적합한 케이스 간의 성능을 비교한다.

Model settings. Feature extraction 단계의 세가지 방법의 경우 모두 Imagenet으로부터 사전 훈련된 가중치가 저장된 기본 모델을 불러와서 사용하였고, 이후 Flatten과 dense layer를 통과하여 1024 개의

Feature extraction	Classification	n_estimators	learning_rate
DenseNet	XGBoost	1000	0.05
	LightGBM	1000	0.05
MobileNetV2	XGBoost	1000	0.08
	LightGBM	1000	0.05
EfficientNetV2	XGBoost	1000	0.08
	LightGBM	1000	0.05

Table 2 Parameter using each case

Feature extraction	Classification	Test accuracy
DenseNet	CNN	0.859
	XGBoost	0.993
	LightGBM	0.992
	Stacking	0.993
MobileNetV2	CNN	0.5
	XGBoost	0.984
	LightGBM	0.985
	Stacking	0.985
EfficientNetV2	CNN	0.355
	XGBoost	0.864
	LightGBM	0.867
	Stacking	0.863

Table 1 Test accuracy in each case

DenseNet End-to-End		Pred	
		No	Yes
True	No	433	8
	Yes	116	325

MobileNetV2 End-to-End		Pred	
		No	Yes
True	No	0	441
	Yes	0	441

EfficientNetV2 End-to-End		Pred	
		No	Yes
True	No	32	409
	Yes	160	281

DenseNet +XGBoost		Pred	
		No	Yes
True	No	438	3
	Yes	3	438

MobileNetV2 +XGBoost		Pred	
		No	Yes
True	No	434	7
	Yes	6	434

EfficientNetV2 +XGBoost		Pred	
		No	Yes
True	No	397	44
	Yes	76	365

DenseNet +LightGBM		Pred	
		No	Yes
True	No	438	3
	Yes	4	437

MobileNetV2 +LightGBM		Pred	
		No	Yes
True	No	435	6
	Yes	7	434

EfficientNetV2 +LightGBM		Pred	
		No	Yes
True	No	396	45
	Yes	72	369

DenseNet +Stacking		Pred	
		No	Yes
True	No	438	3
	Yes	3	438

MobileNetV2 +Stacking		Pred	
		No	Yes
True	No	434	7
	Yes	6	435

EfficientNetV2 +Stacking		Pred	
		No	Yes
True	No	390	51
	Yes	70	371

Table 3 Confusion matrix in each case

output을 출력하도록 하였다. 모든 activation function으로는 ReLU를 사용하였다.

CNN 모델을 이용하여 예측을 진행한 경우 feature extraction을 위해 만든 CNN 구조에 3 개의 dense layer와 2 개의 dropout을 번갈아 가면서 쌓아 마지막은 output이 1 개이고 activation function이 sigmoid인 dense layer를 통과하여 0 에서 1 사이의 확률 값의 형태를 가진 값을 출력하도록 하였다. Batch size의 경우 32 로 지정하였으며, epoch의 경우 모델의 실행시간을 고려하여 DenseNet과 MobileNetV2 는 5, EfficientNetV2 는 1 로 지정하였다. 최적화에는 Adam optimizer를 사용하였다.

XGBoost와 LightGBM을 이용하여 예측을 진행한 경우, 트리 생성 횟수를 의미하는 n_estimators와 각 트리의 학습률을 의미하는 learning_rate를 제외한 모든 파라미터는 default value를 이용하였다. n_estimators와 learning_rate의 경우 몇 가지 값으로 모델링을 시도해본 뒤 좋은 성능을 보이는 값을 이용하였다. 케이스별로 지정한 n_estimators와 learning_rate는 Table 1 과 같다.

4.2. Results

각 case 별로 train set을 이용하여 모델을 만들고 test set에 대해 예측하였을 때 test accuracy와 confusion matrix는 각각 Table 2, Table 3 과 같다.

Feature extraction 방법 별로 모델의 성능을 비교해보면, 먼저 DenseNet의 경우 stacking을 제외한 개별 모델의 test accuracy는 XGBoost가 0.993 으로 가장 높았다. DenseNet-XGBoost의 confusion matrix를 보면 전체 test set 중 6 개의 이미지를 제외하고는 실제 label로 잘 예측한 것을 알 수 있다. Stacking의 결과를 보면 test accuracy가 0.993 으로 XGBoost의 예측 결과와 같은 값으로 높은 정확도를 보임을 알 수 있다.

MobileNetV2 로 feature extraction을 진행한 케이스를 보면, MobileNetV2 로 예측까지 진행했을 때 마스크를 착용한 이미지와 착용하지 않은 이미지 모두 착용한 것으로 예측되었다. 반면 XGBoost와 LightGBM는 약 98%의 test accuracy를 보여 stacking 결과 0.985 의 test accuracy를 얻을 수 있었다.

EfficientNetV2 의 경우, stacking한 결과의 test accuracy가 0.863 으로 앞의 다른 두 feature extraction 방법에 비해서는 정확도가 낮았다. 개별 모델의 성능을 보면, EfficientNetV2 의 accuracy가 0.355 으로 다소 낮은 값을 보여 해당 예측치의 영향을 받아 stacking의 결과가 XGBoost와 LightGBM 단일 모델의 결과보다 좋지 않았던 것으로 예상해볼 수 있다.

3 개의 feature extraction 방법 모두 해당하는 CNN 모델로 마스크 착용 여부에 대해 예측했을 때보다 boosting 방법을 이용하여 예측했을 때의 test accuracy가 더 높은 값을 얻었다. 또한 CNN 모델을 이용한 예측에 비해 stacking 했을 때의 성능이 더 좋아진 것을 확인할 수 있었다.

5. Conclusions

본 연구에서는 CNN과 머신러닝의 boosting 및 stacking 방법을 결합한 모델을 통해 사람의 얼굴 이미지가 주어졌을 때 그 사람의 마스크 착용 여부에 대해 예측하였다. DenseNet을 backbone으로 하여 feature extraction 한 뒤 3 가지 예측방법을 통해 예측치를 얻고 이를 stacking했을 때의 test accuracy가 0.993 으로 가장 높은 정확도를 얻을 수 있었다.

Classification 과정에서 face detection 방법론 중의 하나인 ArcFace를 이용하여 예측을 해보고자 했으나, ArcFace를 구현할 수 있는 별도의 패키지 없어 코드만을 이용하여 구현하는 과정에서 여러 오류를 겪어 ArcFace까지 구현에 어려움을 겪었다. Face detection을 목적으로 만들어진 arcface를 기반으로 classification을 진행했을 때 본 연구 결과보다 딥러닝을 통한 예측에 개선의 여지가 있을 것으로 생각된다.

또한 CNN 모델을 적합하는 시간이 boosting보다 훨씬 오랜 시간이 소요되어 epoch을 크게 지정하지 못했는데, epoch을 늘려 학습 과정을 길게 한다면 성능이 좋아질 수도 있을 것이라고 예상한다.

Boosting의 측면에서는 파라미터의 optimal value를 찾는 방법인 grid search를 통해 optimal parameter를 찾는 방식으로 개별 모델의 성능을 개선해볼 수 있을 이며, 본 연구에서 사용한 3 개의 예측 모델링 외에도 다른 방법을 더 추가하여 3 개보다 많은 예측치를 이용해 개선된 stacking을 시도해볼 수 있을 것이라고 생각된다.

References

[1] F.Schroff, D.Kalenichenko, J.Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. arXiv: 1503.03832, 2015

[2] J.Deng, J.Guo, N.Xue, S.Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. arXiv: 1801.07698, 2018

[3] S.Sethi, M.Kathuria, T.Kaushik. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread. Journal of Biomedical Informatics, Volume 120, 103848, 2021.

[4] D.Montero, M.Nieto, P.Leskovsky, N.Aginako. Boosting Masked Face Recognition with Multi-Task ArcFace. arXiv: 2104.09874, 2021

[5] M.Loey, G.Manogaran, M.Taha, N.Khalifa. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement.2020.108288, 2021

[6] G.Huang, Z.Liu, L.Maaten, K.Weinberger. Densely Connected Convolutional Networks. arXiv: 1608.06993, 2018

[7] M.Sandler, A.Howard, M.Zhu, A.Zhmoginov, L.Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv: 1801.04381, 2018

[8] M.Tan, Q.Le. EfficientNetV2: Smaller Models and Faster Training. arXiv: 2104.00298, 2021

[9] T.Chen, C.Guestrin. XGBoost: A Scalable Tree Boosting System. arXiv: 1603.02754, 2016

[10] G.Ke, Q.Meng, T.Finley, T.Wang, W.Chen, W.Ma, Q.Ye, T.Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30, 2017

[11] Z.Wang, G.Wang, B.Huang, Z.Xiong, Q.Hong, H.Wu, P.Yi, N.Wang, Y.Pei, H.Chen, Y.Miao, Z.Huang, J.Liang. Masked Face Recognition Dataset and Application. arXiv: 2003.09093, 2020