Reconhecimento de emoções humanas através da análise de expressões faciais aplicado a visão computacional

Guilherme Fumagali Marques, Guilherme Silva de Camargo
UFSCar - Universidade Federal de São Carlos
Rodovia João Leme dos Santos, KM 101
guilhermefumagali@estudante.ufscar.br, guilhermecamargo@estudante.ufscar.br

Abstract

O problema abordado no projeto é a identificação de emoções humanas usando visão computacional. Este tema se torna interessante ao pensarmos em relação a quantidade de aplicações práticas possíveis, nesse sentido, propomos utilizar métodos de Aprendizado Profundo para identificar a emoção predominante numa imagem humana, assim, construindo uma API que posteriormente, pode ser aplicada em diversos outros contextos. Nesse sentido, nosso trabalho propõe algumas técnicas a fim de melhorar o desempenho na identificação de oito tipos de emoções diferentes, como a técnica de regularização para aumento de dados, sendo as inversões horizontais. Além disso, foram testadas três métodos de aprendizado supervisionado, Rede Neural Convolucional (CNN) padrão, Rede Neural Convolucional VGG16, e Rede Neural Residual ResNet, tendo sido escolhida por desempenho aliado a tempo de execução a Rede Neural Convolucional (CNN) padrão, baseado na métricas de avaliação acurácia, aliado à melhor desempenho de custo computacional. Por fim, obtivemos desempenhos superiores aos encontrados em outros trabalhos, artigos que propunham identificar 8 emoções distintas, dessa forma, tendo um resultado satisfatório.

1. Introdução

A identificação de emoções humanas é uma área de estudo consolidada principalmente na psicologia, envolvendo estudos das emoções ou inteligência emocional, por exemplo. Por conta disso, têm-se uma ampla variedade de conteúdos acadêmicos e dados relacionados a este tema. Ademais, atualmente encontra-se fotos e gravações em muita abundância, seja por smartphones ou câmeras de seguranças espalhadas por todo o planeta, contribuindo diretamente para um volume enorme de dados, e explicitando a necessidade de maiores utilizações dos mesmos. Assim, aplicado à visão computacional, projetos com o intuito de serem acessíveis aos usuários podem encontrar grande mercado, tal como aplicativos de celular e plugins de navegadores, que podem utilizar esses estudos da psicologia para classificar a emoção mais presente em uma imagem.

Com isso, este trabalho propõe um modelo de rede neural convolucional para classificar emoções mediante imagens, com a escala de emoções definida pela teoria emocional propostas por Paul Ekman e Wallace V. Friesen que identifica sete emoções básicas universais: raiva, nojo, medo, desprezo, surpresa, tristeza e alegria. Estas emoções são caracterizadas por expressões faciais universais, sugerindo que são inatas e não culturalmente determinadas.

Por conta de ter sido amplamente aplicada em vários campos da psicologia, a Escala de Emoções de Ekman e Wallace é amplamente aceita na comunidade científica como uma teoria robusta e confiável na compreensão das emoções humanas, desse modo, existe a segurança de captar as principais e necessárias emoções provenientes de seres humanos. No entanto, no contexto problema de classificação presente, também é definido como classe as imagens com expressões neutras, que enquadra todas as outras expressões não identificadas pela teoria.

2. Trabalhos Relacionados

Como citado anteriormente, essa é uma área de pesquisa movimentada no âmbito acadêmico, o que reflete em diversos artigos sobre o tema. No entanto, com o intuito de comparação, será selecionado trabalhos que usam os mesmos dados para a pesquisa, assim trazendo a possibilidade de medir as diferenças das abordagens.

No contexto do dataset utilizado neste trabalho, ele tem sido amplamente utilizado na comunidade de visão computacional para o reconhecimento de emoções em imagens. Os trabalhos relacionados foram publicados com o objetivo de melhorar o desempenho do reconhecimento de emoções no AffectNet usando diferentes abordagens e arquiteturas de rede neural.

Utilizando diferentes redes neurais, o trabalho proposto

em "Multi-head Cross Attention Network for Facial Expression Recognition", utilizou da base de dados AffectNet para avaliar o novo modelo de rede neural proposto no artigo. Essa rede atingiu cerca de 62% de acurácia na base de dados, o que já foi melhor do que outras oito arquiteturas de redes neurais citadas no artigo, utilizando de quatro mil imagens para teste.

O artigo "Using Self-Supervised Auxiliary Tasks to Improve Fine-Grained Facial Representation" propõe o uso de modelos híbridos de aprendizado para reconhecimento das emoções disponíveis na versão de oito classes da base de dados AffectNet. Esses modelos propostos também atingiram por volta de 62% de acurácia no conjunto de validação.

Por fim, o artigo "Expression, Affect, Action Unit Recognition: Aff-Wild2, Multi-Task Learning and ArcFace" foca no reconhecimento de expressões faciais utilizando como dados para treinamento o dataset AffectNet, também com oito emoções. Com a métrica F1-score obteve-se o melhor resultado de 58% utilizando AlexNet.

Em suma, esses artigos conseguem atingir um grande leque de implementações, e nelas os resultados são usualmente próximos a 60% de acurácia. Baseando-se nisso, este trabalho propõe uma implementação utilizando de técnicas de pré-processamento, para que obtenha resultados superiores.

3. Dados

O projeto contém um conjunto de dados de expressões faciais rotuladas com 8 emoções definidas na teoria emocional de Paul Ekman. A princípio, os dados foram coletados de dois datasets públicos, AffectNet e Fer 2013, porém foi decidido manter somente o AffectNet, visto que é um dos maiores bancos de dados de imagens faciais de expressões emocionais disponíveis, e portanto iria conter a maioria das imagens, sendo que a qualidade dessas imagens são superiores às do Fer 2013. Com isso, aliado aos pré-processamentos que abordaremos a frente que aumentarão a quantidade de imagens, decidiu-se manter somente o dataset AffectNet.

Assim, explicitando mais o conjunto de dados escolhido, temos que com o total de 291650 imagens retiradas da internet, o conjunto de dados AffectNet foi criado pela Universidade de Pittsburgh em colaboração com a empresa de tecnologia Affectiva, usando uma combinação de técnicas automatizadas e análise humana para rotular as emoções das pessoas nas imagens. As imagens foram coletadas de fontes públicas, incluindo a internet, e incluem uma ampla variedade de idades, gêneros e etnias. A distribuição das classes neste dataset pode ser visualizado na figura 1. Nela, é possível visualizar uma certa discrepância entre as classes, com quase 3/4 das imagens da classe Neutra e Feliz.

Na sequência, iniciando os pré-processamentos, converteu-se todas imagens em escala de cinza, pois o

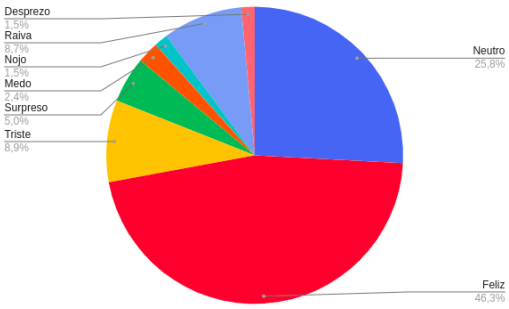

Figure 1. Distribuição das classes do AffectNet

dataset inclui tanto imagens nessa escala, como imagens em RGB. A importância de utilizar essa técnica está na consequência da redução da dimensionalidade (3 canais RGB para 1 em escala de cinza), o que impacta diretamente na quantidade de dados para o modelo processar, eliminando a dificuldade em lidar com a variação de cores das imagens, considerando que a informação sobre a cor não tem tanta correlação com os padrões faciais das emoções. Além disso, eliminar as cores das imagens tem como consequência também reduzir o viés relacionado às características físicas e raciais do ser humano que pode problematizar o modelo.

Além disso, foi utilizado desfoque gaussiano que suaviza as imagens aplicando uma função Gaussiana com o intuito de remover ruído das imagens, esse desfoque foi aplicado com parâmetros baixos, inferindo uma mudança discreta nas imagens, visto que estas já tem uma resolução baixa. Ademais, utilizamos o redimensionamento das imagens para 48x48 a fim de minimizar o tempo de processamento.

Por fim, foi realizado a equalização do histograma, técnica comumente usada que modifica a distribuição de intensidades de pixels, tornando-a mais uniforme, o que pode ajudar na padronização das amostras e na melhora da qualidade delas para o modelo.

4. Métodos

O pipeline da abordagem a ser realizada consiste em aplicar os pré-processamentos necessários, treinar alguns modelos e compará-los de modo a escolher o modelo com as melhores medidas de desempenho.

Como citado no pré-processamento, realizou-se com o intuito de aumentar os dados, técnicas de regularização em que inversões horizontais aplicadas nas imagens originais, como exemplificado na figura 3, invertem a imagem, obtendo-se o dobro da quantidade de imagens inicial. Assim a base de dados a ser passada para treinamento dos modelos contém muito mais dados, o que representou um ganho nas métricas em relação a testes feitos com o dataset

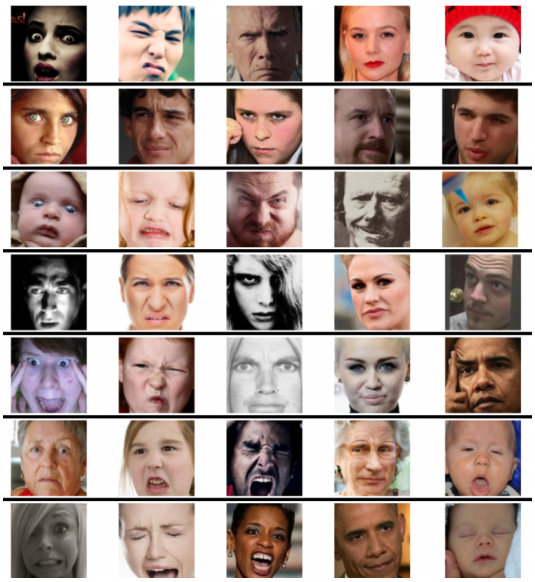

Figure 2. Exemplos de imagens do dataset AffectNet

Classe	Rótulo	Qtd. de amostras
1	Neutro	150748
2	Feliz	269830
3	Triste	51918
4	Surpreso	29180
5	Medo	13756
6	Nojo	8606
7	Raiva	50764
8	Desprezo	8498

Table 1. Distribuição dos dados

Épocas	Filtro (1,1)	Filtro (3, 3)	Filtro (5, 5)
1	0,6910	0,7022	0,6660
2	0,7031	0,7129	0,6882
3	0,7101	0,7180	0,7001
4	0,7161	0,7210	0,7092
5	0,7202	0,7221	0,7110

Table 2. Precisão para diferentes filtros com CNN

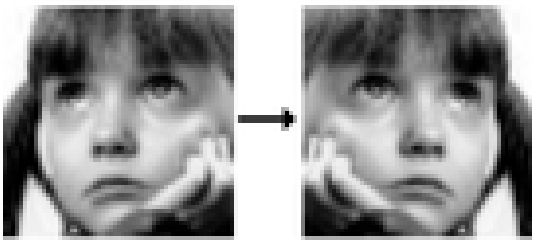

Figure 3. Exemplo de aplicação da técnica de inversões horizontais

sem as imagens invertidas.

Ao fim, como mostrado na tabela 1, obteve-se como resultado quantidades na ordem de 500 mil imagens, após utilizar a técnica de regularização, para treinamento do modelo, mas ainda mantendo a distribuição original do dataset.

Por fim, sobre os modelos a serem testados, além da Rede Neural Convolucional citada, as arquiteturas de Rede ResNet50 e VCC16 também foram treinadas. Porém abordaremos os desempenhos gerais na parte de experimentos, elucidando neste momento apenas o funcionamento de cada rede neural, porém terá um enfoque maior na CNN contemplando diferentes hiperparâmetros, pois como será visto a frente essa teve desempenho mais satisfatório sendo maior explorada.

4.1. Rede Neural Convulacional

As redes neurais convolucionais são capazes de aprender representações de nível mais alto de características das imagens, o que lhes permite extrair padrões e características complexas, como bordas, texturas e formas.

Assim como a maioria dos modelos de Aprendizado de Máquina, existem hiperparâmetros que precisam ser ajustados de modo obter melhores resultados no problema em questão. No caso, foi testado diferentes tamanhos de filtros para as camadas convolucionais 2d. Dessa forma, com os resultados obtidos, que podem ser visualizados na tabela 2, temos que o filtro de tamanho (3, 3) obteve os melhores resultados, e portanto, foi selecionado para o modelo.

Nesse sentido, a arquitetura usada nesse projeto é composta por várias camadas, que são organizadas da seguinte forma hierárquica:

A entrada da rede é uma camada Conv2D com 32 filtros de tamanho (3,3) e a função de ativação 'relu'. Esta camada recebe uma entrada de imagens de tamanho (48, 48, 1), onde 1 representa a imagem em escala de cinza. Na sequência, os dados de saída são passados a uma camada MaxPooling2D com um tamanho de pool de (2,2), que reduz a dimensão espacial das saídas da camada anterior.

Ademais, a terceira camada é uma camada convolucional Conv2D com 64 filtros de tamanho (3, 3) e função de ativação ReLU. Esta camada segue a mesma estrutura da primeira camada, mas com o dobro de filtros. Isso permite que a rede aprenda características mais complexas e abstratas das imagens.

Na sequência, a quarta camada é uma camada de pooling MaxPooling2D que reduz a resolução da imagem pela

metade. Ela segue o mesmo padrão da segunda camada, mas com a diferença de que agora a entrada é a saída da terceira camada convolucional. Essa camada reduz ainda mais a dimensão espacial da representação da imagem, mantendo apenas as informações mais importantes e relevantes aprendidas nas camadas anteriores. Essa redução da dimensão espacial torna o treinamento da rede mais rápido e eficiente e ajuda a evitar o overfitting.

Outrossim, a quinta camada é igual a terceira camada, com a diferença que a entrada é a saída da quarta camada de pooling MaxPooling2D. Como essa camada reduz a resolução da imagem pela metade, a entrada para a quinta camada é uma imagem menor em relação à entrada original. Isso significa que a quinta camada tem que aprender características mais abstratas e complexas da imagem para conseguir fazer as previsões corretas.

Enfim, a próxima camada é uma camada Flatten, que transforma a saída da camada convolucional em um vetor unidimensional, para que possa ser processado pelas camadas totalmente conectadas que vêm a seguir. Além disso, a sétima camada é uma camada Dense com 64 neurônios e a função de ativação 'relu'. Por fim, a oitava e última camada é uma camada Dense com 8 neurônios e a função de ativação 'softmax', usada para classificar as imagens em uma das 8 classes possíveis.

4.2. ResNet50

No projeto foi utilizado a ResNet em sua versão com 50 camadas, que são projetadas para classificação de imagens. A única modificação na arquitetura da rede foi que a camada de saída original da rede não foi incluída no modelo, pois foi adicionada novas camadas de saída mais tarde, para se enquadrar com as classes propostas nesse problema de classificação. Ademais, a ResNet50 é inicializada com pesos não treinados.

A entrada da rede é uma imagem em escala de cinza de 48x48 pixels, que é processada por uma série de camadas convolucionais, seguida por camadas de pooling e camadas totalmente conectadas (Dense). Ademais, a camada Dense final tem 8 neurônios e usa a função de ativação softmax para produzir uma distribuição de probabilidade sobre as possíveis classes de saída.

Além disso, a função de perda usada durante o treinamento é a 'sparse_categorical_crossentropy', que é uma função de perda comum para problemas de classificação.

Com isso, após definida a arquitetura da rede, o código compila a rede usando o otimizador Adam, que é uma variação do gradiente descendente estocástico que adapta a taxa de aprendizado ao longo do treinamento, a função de perda e a métrica de avaliação definidas anteriormente.

Épocas	ResNet	Conv2D	VGG
1	0,6564	0,7022	0,6614
2	0,6801	0,7129	0,6892
3	0,7026	0,7180	0,6956
4	0,7145	0,7210	0,7148
5	0,7298	0,7221	0,7236

Table 3. Precisão para diferentes modelos

4.3. VGG16

A VGG16 é uma das arquiteturas de CNN mais populares e influentes da última década, tendo sido amplamente utilizada em aplicações de visão computacional e aprendizado profundo. A arquitetura VGG16 consiste em 13 camadas convolucionais e 3 camadas densas, com um total de 138 milhões de parâmetros treináveis. A camada convolucional consiste em uma sequência de filtros de convolução com tamanho de 3x3 pixels e camadas de pooling com tamanho 2x2 pixels.

A rede neural utilizada no projeto é uma implementação da VGG16, que além da arquitetura proposta por esse modelo de rede, conta também com outras camadas para se adaptar ao problema de classificação do projeto. Dessa forma, essa rede neural consiste em quatro camadas, assim como a ResNet50. A primeira camada é a VGG16, que é uma camada de convolução com pesos inicializados de forma aleatória. Além disso, a VGG16 inclui todas as camadas de convolução e pooling necessárias para extrair recursos da imagem de entrada. Depois disso, temos as duas últimas camadas que são necessárias para termos as saídas correspondentes às oito classes do dataset AffectNet.

5. Experimentos

Nesta seção serão abordados os experimentos realizados para definir-se o melhor modelo para o problema, baseado em medidas descritivas e no tempo de execução, além de medir o ganho relacionado a técnica de regularização no modelo escolhido.

O conjunto de validação a ser utilizado para critério de desempenho dos modelos foi gerado a partir de uma separação holdout estratificada. Foi-se separado 10

Os resultados das acurácias dos modelos podem ser visualizados na tabela 3. Nela, pode-se observar os resultados mais relevantes no contexto da Rede Neural Convolucional Padrão, visto que, mesmo com resultados parecidos com a ResNet, essa rede se sai melhor devido ao seu custo computacional reduzido, levando cerca de 40 vezes menos tempo e energia para ser treinada. Por outro lado, a rede VGG apresentou resultados insatisfatórios tanto nas métricas de acurácia como em relação ao custo computacional envolvido no processo de treinamento.

Com isso, tendo o melhor modelo selecionado, é apre-

Épocas	S/ regularização	C/ regularização	Ganho
1	0,6878	0,7022	1,02
2	0,7021	0,7129	0,98
3	0,7077	0,7180	0,98
4	0,7089	0,7210	0,98
5	0,7155	0,7221	0,99

Table 4. Precisão utilizando CNN Padrão

sentado os ganhos referentes às técnicas de regularização, esta que pode ser visualizada na tabela 4. Infere-se que essa regularização trouxe ganhos relativamente significativos, considerando que o processo de treinamento dessa rede não é muito custoso, e ainda obteve-se praticamente 1% a mais de acurácia no mesmo conjunto de validação.

Conclui-se com os resultados dos experimentos aliados ao intuito inicial do projeto, o qual pretendia-se desenvolver uma aplicação de fácil acesso, para tanto necessitando de métodos menos custosos no âmbito computacional, evidenciou-se que a Rede Neural Convolucional Padrão (CNN) é de fato a melhor opção, pois não apresentou desempenho inferior evidente em termos de acurácia, saindo-se melhor que um dos métodos aplicados, e exige um custo computacional menor na ordem de 6 vezes pela experiência obtida nos experimentos.

6. Aplicação

Inicialmente discutiu-se qual seria a aplicação final desenvolvida, optou-se por uma API, visto que essa pode ser usada em diversas outras aplicações, e assim, abranger maiores contextos de identificação de expressões faciais. Dessa forma, como o principal apêndice gerado do projeto, foi desenvolvido uma aplicação de fácil acesso com o intuito de tornar acessível a utilidade da rede neural treinada.

Portanto, essa aplicação está hospedada como um serviço WEB, e baseia-se na arquitetura cliente-servidor. No contexto, temos um servidor que recebe uma imagem sem nenhum tipo de pré-processamento, e retorna a expressão facial presente na imagem, se houver. Ademais, do lado do cliente, foi projetada uma interface que possibilita ao usuário fazer o upload da imagem e receber a resposta do servidor.

Além disso, toda arquitetura pode ser visualizada na figura 4, onde o servidor é responsável por realizar todos os processamentos necessários, possibilitando que o cliente acesse o serviço até mesmo em dispositivos móveis.

Para a detecção de faces, foi utilizado o classificador de cascata de Haar, que é um algoritmo de Aprendizado de Máquina de detecção de objetos baseado em características de imagem. Semelhante a uma rede neural convencional, o algoritmo funciona através da análise de características locais da imagem em cascata que são aprendidas através de

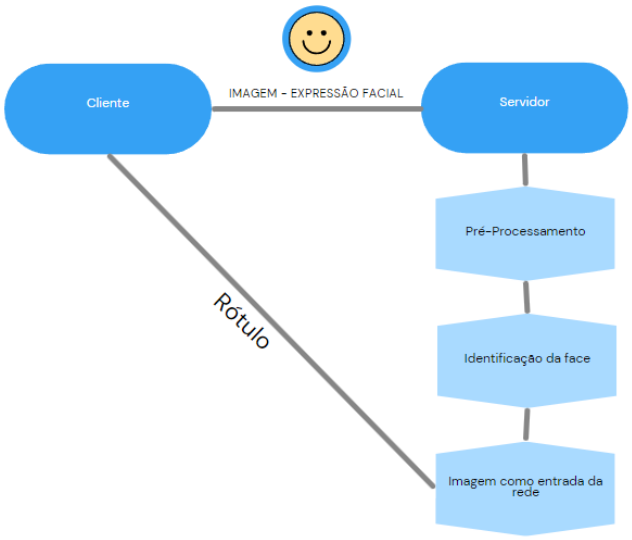

Figure 4. Arquitetura da Aplicação

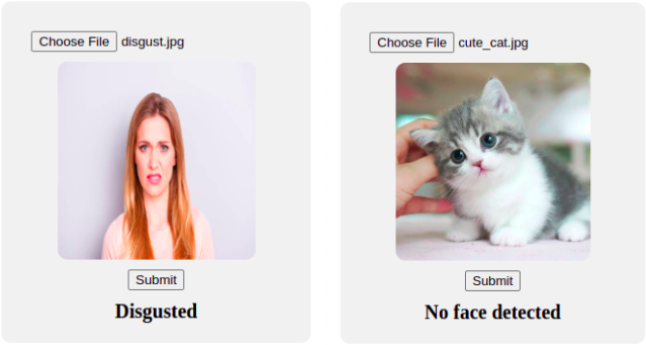

Figure 5. Exemplo de utilização da API

imagens de exemplos. No âmbito de detecção de rostos, o classificador de cascata de Haar utiliza um conjunto de características específicas para detectar as regiões do rosto, como olhos, nariz, boca, etc. Quando várias dessas características são encontradas próximas umas das outras, o algoritmo conclui que um rosto foi detectado naquela região.

Por fim, tornando o servidor público, desenvolvedores podem utilizar desse ponto de acesso ao servidor para criar suas próprias aplicações, como por exemplo um plugin de navegador.

7. Conclusão

Infere-se, portanto, que o principal problema abordado na introdução: acessibilidade da aplicação foi resolvido, visto que a API permite a utilização do reconhecimento de expressões faciais em diversos contextos. Aliado ao fator de apresentar resultados iguais ou superiores a trabalhos relacionados, pode-se concluir que os métodos apresentados são de fato um bom resultado. Além disso, a aplicação

conta com uma utilização simplificada, o que permite um uso mais difundido.

Ademais, a aplicação tem a limitação de identificar apenas um rosto por vez, não sendo possível utilizá-la para identificar vários rostos em uma só imagem, mas isto pode ser modificado, abrindo perspectiva de melhoria para a aplicação em novas versões futuras. Dessa forma, o projeto mostra-se satisfatório ao realizar as principais demandas esperadas.

References

[1] Wen, Zhengyao, et al. "Distract your attention: Multi-head cross attention network for facial expression recognition." arXiv preprint arXiv:2109.07270 (2021).

[2] Kollias, Dimitrios, and Stefanos Zafeiriou. "Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface." arXiv preprint arXiv:1910.04855 (2019).

[3] Pourmirzaei, Mahdi, Gholam Ali Montazer, and Farzaneh Esmaili. "Using self-supervised auxiliary tasks to improve fine-grained facial representation." arXiv preprint arXiv:2105.06421 (2021).