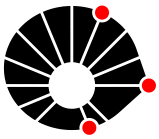
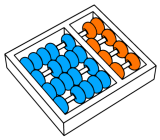




Aumento semântico de dados com Modelos Generativos para Classificação de Vídeos

Francisco Carneiro *Adín Ramírez Rivera*

Relatório Técnico - IC-PFG-19-21
Projeto Final de Graduação
2019 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Aumento semântico de dados com Modelos Generativos para Classificação de Vídeos

Francisco Carneiro* Adín Ramírez Rivera†

2019 - Julho

Resumo

O objetivo principal deste projeto foi explorar técnicas de classificação de vídeos para avaliar e qualificar dados sintéticos criados a partir de Modelos Generativos. Para isso, explora-se quatro modelos para classificação: MobileNet [1], Resnet50 [3] e um modelo de CNN proposta, com o intuito de validar dois cenários de dados. Um cenário com os dados originais e o segundo cenário com o acréscimo de dados sintéticos necessários para balancear os datasets. Assim, pode-se notar a influência dos dados sintéticos na classificação obtendo ganhos de até 6 % em acurácia.

1 Introdução

A evolução dos meios digitais de comunicação juntamente com o advento e popularização da internet trouxe uma grande disponibilidade de dados. Em paralelo a isso, o desenvolvimento de hardware, como CPU (Central Processing Unit) e GPU (Graphic Processing Unit) possibilitou o desenvolvimento de tecnologias relacionadas a processamento de dados, dentre elas, a Inteligência Artificial.

Modelos inteligentes, como Redes Neurais foram se aperfeiçoando ao do tempo e assim como sua complexidade. Chegando nas Redes Neurais profundas [4] na qual uma quantidade exacerbada de dados é necessária para que estes métodos possam produzir resultados esperados (convergir). Para isso, há uma vertente de estudos que trabalha na criação de métodos que possam gerar novas amostras para treinamento e teste, que são as Modelos generativos profundos.

Redes neurais generativas adversariais (GAN) [5] introduzidas em 2014 podem ter grandes funcionalidades, detecção de anomalias, criação de dados sintéticos como imagens e voz, por exemplo. Seu aprendizado se dá de forma sem supervisão, na qual os dados não contém rótulos, e contém dois componentes principais: Gerador e um Discriminador. Em suma, o Gerador é responsável por gerar dados sintéticos a partir de a partir de ruído proveniente de uma distribuição de probabilidade conhecida e o Discriminador é responsável por diferenciar os dados sintéticos com os dados reais.

*Instituto de Computação - Unicamp
†Instituto de Computação - Unicamp

Uma forma de avaliar a qualidade dos dados gerados é observar a influência destes em algum processo de classificação.Ou seja, dada a eficiência de uma série de classificadores com os dados originais compará-la com a eficiência destes classificadores (com os mesmos parâmetros) com os dados sintéticos. Então, pode-se observar se estes dados podem ser utilizados para obter melhores resultados e consequentemente melhores modelos.

2 Trabalhos Relacionados

Em termos de classificação de vídeo, existem muitos trabalhos publicados, um deles é Beyond Short Snippets: Deep Networks for Video Classification [14] na qual aborda o uso de redes neurais convolucionais [8] e redes neurais recorrentes [15], com uma arquitetura profunda para combinar as imagens (frames) do vídeo de forma sequencial. Utilizou-se a combinação de uma rede neural convolucional (CNN) e uma rede neural recorrente (LSTM), ligando a saída da CNN na entrada da LSTM. Assim, utilizando dos datasets: Sports 1 milhão, com 73,1 % de acurácia, UFC-101 com 88,6 %.
Em sequência, outro trabalho importante, Large-Scale Video Classification with Convolutional Neural Networks [16], que utilizou uma CNN na classificação de vídeos em grande escala, utilizando uma forma de estender a conectividade de uma CNN no domínio do tempo para aproveitar as informações temporais dos vídeos. Foi testado no dataset composto por 1 milhão de vídeos do Youtube subdivididos em 487 classes, obtendo 63,9 % de acurácia.
Para a geração de dados sintéticos para o aumento semântico de dados, o artigo: Generative Adversarial Networks [5] propôs uma nova estrutura,subdividida em um modelo generativo é capaz de capturar a distribuição de dados e um modelo discriminativo capaz de estimar a probabilidade de uma amostra ter vindo dos dados de treinamento em vez do modelo generativo. A forma de treinamento foi dada na forma de que o modelo generativo maximiza a probabilidade do modelo discriminativo errar. Ambos os modelos são redes neurais, compostas por perceptrons em camadas e retropropagação no processo de treinamento.
Descobriu-se, no artigo: Least Squares Generative Adversarial Networks [17], o discriminador, como um classificador com a função de perda *cross entropy*, que a essa abordagem pode levar ao problema de perda de gradientes durante o processo de aprendizagem, ou seja, não chegando a um modelo convergente. Para consertar, o artigo propôs que as redes generativas adversariais com mínimos Quadrados (LSGANs [17]) que adotam a função de perda de mínimos quadrados ao invés de *cross entropy*, com o intuito de reduzir a perda de gradiente (divergência). Comparou-se as GAN's [5] com as LSGANs em duas bases de dados: LSUN e CIFAR-10.
Em GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification [15] explora-se utilização de Generative Adversarial Networks (GAN's) para a criação de dados sintéticos médicos para que possa melhorar os resultados obtidos na classificação. O banco de dados contém 182 imagens de lesões hepáticas dividindo-se em três classes: Cistos, Hemangioma e metástase. Assim, utilizou-se métodos de geração de dados sintéticos para aumentar significativamente o número de amostras e assim aumentar a acurácia e a qualidade do classificador. O resultado da classificação somente com dados reais foi uma acurácia de 78.6 %. Após a inserção de dados sintéticos, o

resultado foi de: 85.7 %
Data Augmentation Generative Adversarial Network [14] aborda-se o ganho em classificação utilizando dados sintéticos criados a partir de modelos generativos. Para isso, utilizou-se três datasets: Omniglot, VGGFaces e EMNIST. O EMNIST é um dataset composto por digitos em cada imagem, composto por 814.255 amostras divididas em 62 classes. Já o Omniglot, é um dataset composto por 1600 imagens contendo caracteres escritos a mão divididos em 50 classes. E por fim, o VGGFaces é um dataset composto por 2.6 milhões de imagem divididos em 2622 classes. A partir disso, os resultados obtidos foram para experimentos com poucos dados: Ganho de 13 % para Omniglot (de 69% para 82 %), de 2.1 % para o EMNIST (de 73.9 % para 76 %) e por fim, 7.5 % para VCCFaces (de 4.5 % para 12 %).

3 Datasets

Para os experimentos para a classificação de datasets com e sem dados sintéticos gerados por redes neurais adversariais, utilizou-se dois principais datasets. O Cohn-Kanade [6] dataset e MUG [7] (Multimedia Understanding Group) baseados em expressões faciais.

3.1 Cohn-Kanade - Dataset

O The Cohn-Kanade AU-Coded Facial Expression é um dataset composto por vídeos que representam expressões faciais com 123 indivíduos (Classes de aparência) e 7 classes de emoções, são estas: Tristeza, surpresa, felicidade, medo, raiva, desprezo e desgosto. A distribuição de dados (Figura 1) mostra que o dataset é desbalanceado, tento as classes Desprezo, Medo e Tristeza com uma quantidade de dados consideravelmente menor que as outras, o que, pode ser problemático no processo de classificação.

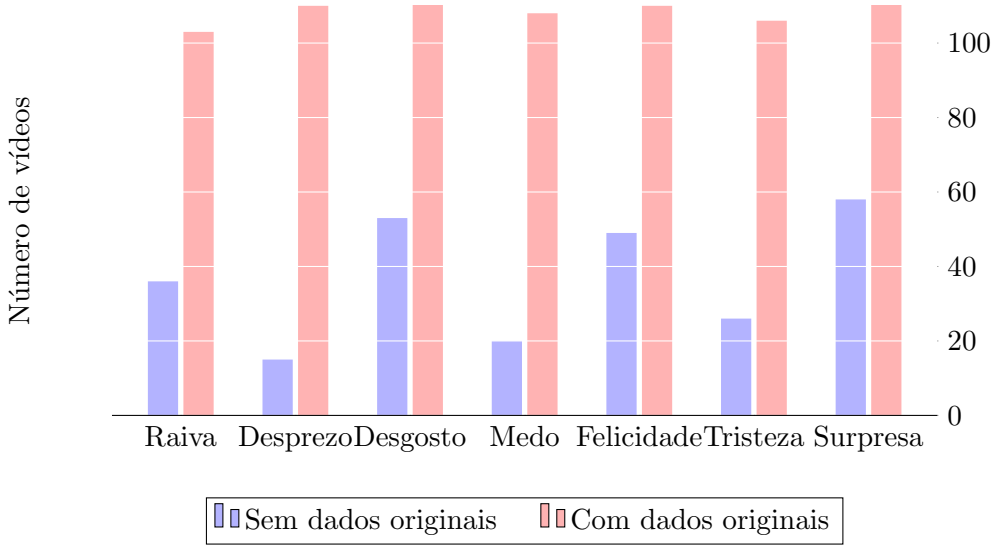


Figura 1: Distribuição de Dados do dataset CK

Para os experimentos com o aumento de dados, utilizou-se os dados sintéticos para que o dataset pudesse ficar minimamente balanceados. Pode-se observar que, a distribuição de dados mostra um dataset minimamente balanceado quando há a presença de dados criados, que pode ajudar diretamente no processo de classificação e também um aumento considerável no número de amostras. Seguem exemplos de dados gerados e dados sintéticos gerados para o respectivo dataset:

(a) Classe Raiva Original (b) Classe Raiva Sintético

(c) Classe Felicidade Original (d) Classe Felicidade Sintético

(e) Classe Tristeza Original (f) Classe Tristeza Sintético

Figura 2: Exemplos de dados originais e sintéticos para o dataset CK.

3.2 Multimedia Understanding Group - Dataset

MUG - Multimedia Understanding Group dataset, é um banco de dados que consiste em vídeos de 86 indivíduos compostos por expressões faciais. Compõe os indivíduos 35 mulheres e 51 homens, todos de origem caucasianos. Para a criação dos dados sintéticos e para o processo de classificação considerou 6 classes principais, sendo elas: Surpreso, Tristeza, Medo, Raiva, Desgosto, Felicidade. Assim a distribuição do dataset ficou:

Gráfico 2: Distribuição de Dados do dataset MUG.

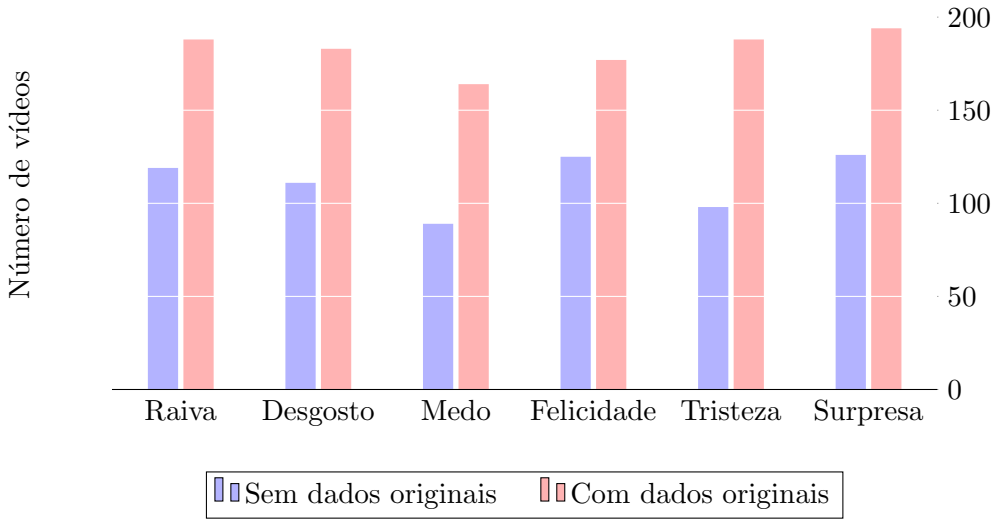


Figura 3: Distribuição de Dados do dataset MUG.

Os dados sintéticos foram criados para poder balancear minimamente o dataset, de forma que pode-se observar sua influência direta na classificação. E também, que em relação ao dataset CK [6] o balanceamento do dataset MUG [7] é mais consistente, ou seja, precisou-se de menos vídeos, proporcionalmente, para a balanceado dos dados, e portanto, a classificação dos CK [6] pode ter influência direta, já que há mais dados sintéticos do que originais no banco de dados balanceado e, como visto na Figura 2, há uma maior presença de ruído nos dados criados.

Segue, a Figura 4 que trás exemplos do dataset MUG [7].

(a) Classe Raiva Original (b) Classe Raiva Sintético

(c) Classe Felicidade Original (d) Classe Felicidade Sintético

(e) Classe Tristeza Original (f) Classe Tristeza Sintético

Figura 4: Exemplos de dados originais e sintéticos para o dataset MUG.

4 Metodologia

Para este projeto, foi realizada uma série de testes com os classificadores e com os datasets citados.
A ideia principal é criar mecanismos comparativos para que possa-se analisar a influência de dados sintéticos no processo de classificação e consequentemente, poder avaliar a qualidade dos dados criados. Para isso, a partir dos datasets MUG [7] e CK [6], realizou-se o processo de classificação destes, utilizando labels de emoções.

SGD - Stochastic gradient descent e Learning Rate e Decay:
Em suma, o SGD [9] é um método iterativo que tem objetivo de otimizar uma função objetiva. Ele utiliza amostras selecionadas aleatoriamente ou embaralhadas para avaliar os valores dos gradientes, ou seja, tornando a descida do gradiente (otimizando a função até um ponto minimal) melhor e mais precisa no ajuste dos pesos de uma rede neural.

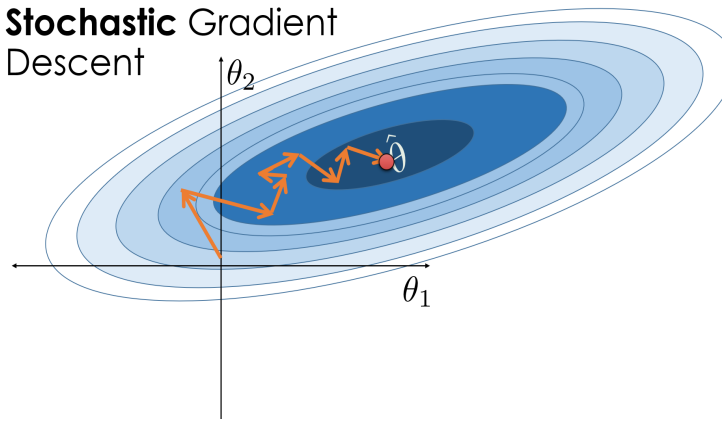


Figura 5: Exemplo de Descida do Gradiente.

Já o *Learning Rate* é um hiperparâmetro responsável por controlar as taxas de atualização dos pesos da rede em relação aos gradientes de perda. Ou seja, quanto menor o valor, mais devagar será o processo de encontrar um ponto minimal no espaço de aprendizagem. É um parâmetro que necessita de um ajuste fino sempre, já que, embora um valor baixo pode levar a um mínimo, pode ser que a convergência de uma rede neural pode demorar muito ou até mesmo ficar preso em uma região que não há um mínimo. A cada iteração o valor do Learning rate é atualizado para que possa evitar o processo ficar preso em uma depressão do espaço de aprendizado e nunca atingir um mínimo local, esse valor é o Decay, ou seja, a taxa de atualização do learning rate a cada iteração.

Droput, Regularização e Batch Normalization:
Dropout [11] é uma técnica que tem por objetivo principal reduzir sobre aprendizado(overfitting) em redes neurais. É feito desligando conexões entre neurônios entre camadas a fim de evitar que o um modelo se especialize no banco de dados.

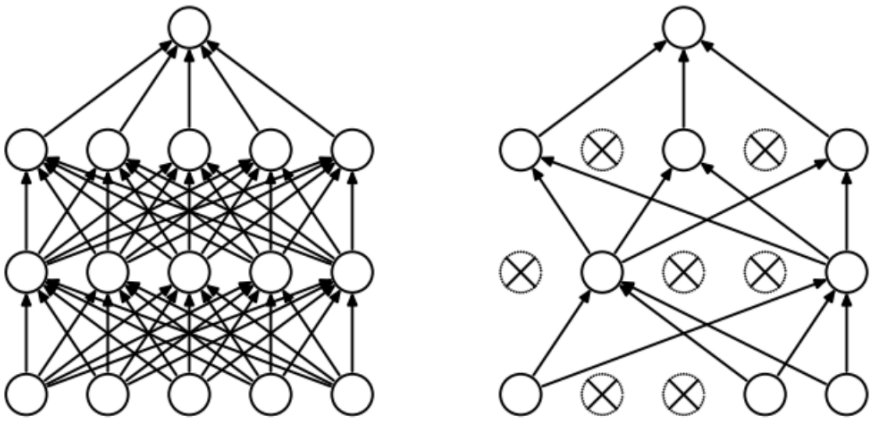


Figura 6: Exemplo de Dropout.

Regularização L2 [10] também é uma forma de evitar sobre aprendizado (overfitting). Trata-se de uma forma de modificar o valor dos pesos a forma a punir pesos que estão fora do esperado (outliers), evitando que estes possam influenciar diretamente no processo de classificação e gerando um modelo super especializado (overfitting).

Batch Normalization [12], diferente daquilo visto neste tópico, é uma técnica que tem o intuito de ajudar na velocidade desempenho e estabilidade de redes neurais. É usada para normalizar a saída de uma camada (entrada da camada subsequente) para que possa melhorar no processo de ativação.

A partir dessas configurações, os modelos foram treinados nos dois cenários: Com e sem dados sintéticos. Na próxima seção, explora-se os resultados obtidos.

5 Modelo Generativo e Classificadores

Para a geração dos dados sintéticos utilizou-se modelos generativos [5] e com os dados gerados, para a classificação, utilizou-se três principais classificadores: MobileNet [1], Resnet [3] e uma CNN proposta para lidar com a pequena quantidade de dados do dataset CK [6].

5.1 GAN: Generative Adversarial Networks

Em redes neurais generativas adversariais [5] (GAN), existem duas redes neurais que compõem todo o modelo generativo, exemplo: Figura 7. A primeira é o gerador. Ela é responsável por criar dos dados sintéticos a partir da distribuição de dados do dataset original. A segunda rede é conhecida como discriminador, ela é responsável por diferenciar os dados criados pelo gerador com os dados da distribuição original. Ou seja, o gerador é responsável por criar um dado que não existe no dataset original e o discriminador é responsável por diferenciar este dado inexistente dos dados reais.

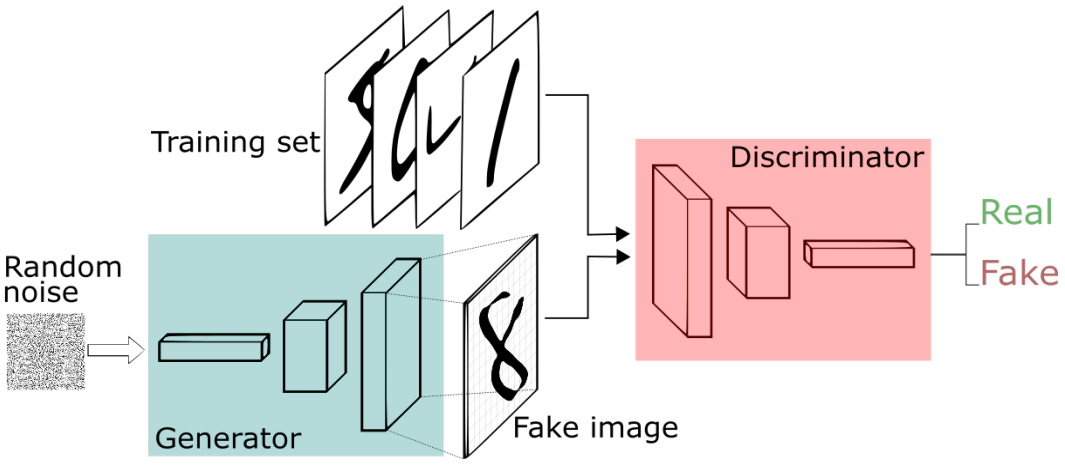


Figura 7: Esquema de uma GAN [5] genérica.

Para a geração de vídeos, há a necessidade de outros modelos para a obtenção do vídeo completo. Assim, o gerador se subdivide em três partes:

Uma das partes é responsável pela captação das características de aparência dos dados reais, como por exemplo: textura, distribuição de cor, ruídos e bordas.

A segunda é responsável pela captação das informações relacionadas a temporalidade do vídeo, obtendo informações do movimento, como fluxo, por exemplo.

E por fim, a terceira é responsável por combinar esses dados e gerar um novo vídeo.

Exemplo: Dado dois vídeos A e B: Passa-se o vídeo A pela rede neural que retira as informações de aparência. Passa-se o vídeo B pela rede neural que retira as informações de movimento. A terceira rede neural é responsável por combinar os dados obtidos pelas duas redes e gerar um novo vídeo C. No final do processo, os vídeos gerados são passados pelo discriminador para diferenciá-los dos dados reais.

Agora, o modelo generativo utilizado no processo de criação dos dados sintéticos é conhecido como Monkey-Net [18], uma nova estrutura de aprendizagem profunda para animação de imagens. Dada uma imagem de entrada com um objeto alvo, no caso deste projeto o objeto alvo de ambos os datasets são indivíduos (pessoas), e uma sequência, um vídeo, que define um movimento de um objeto. O framework é responsável por gerar um novo vídeo em que o objeto alvo é animado de acordo com a sequência apresentada no vídeo de entrada. O movimento é transferido a partir da detecção auto-supervisionada de pontos de interesse (keypoints) tanto no vídeo (que contém o movimento) quanto na imagem que contém o objeto alvo e também seu emparelhamento (o matching) para aplicar as deformações necessárias na imagem com o objeto alvo para cada frame do vídeo de movimento. E assim, adicionalmente, usa-se uma GAN para melhorar o realismo da sequência gerada, ou seja, diminuir a quantidade de ruído produzida. A partir de todo esse pipeline, gera-se um vídeo final que contém a aparência da imagem de entrada com os movimentos do vídeo de saída. A arquitetura utilizada pode ser vista abaixo:

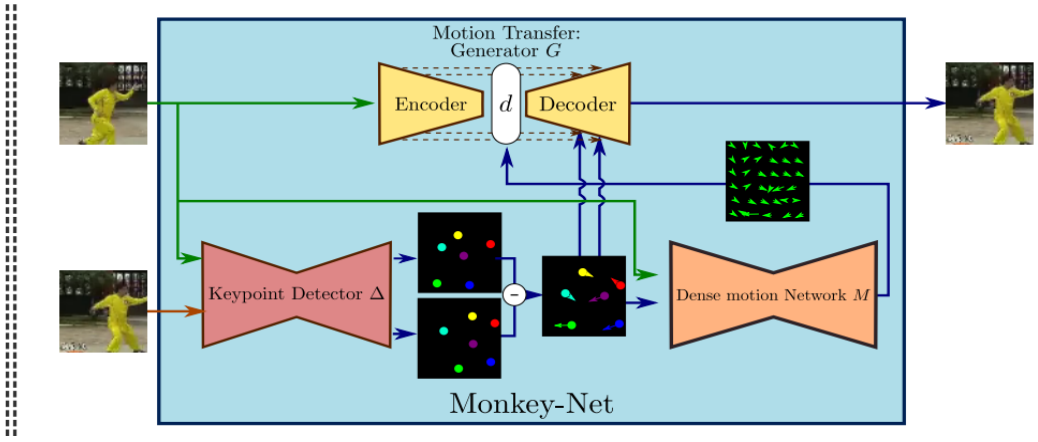


Figura 8: Arquitetura da Monkeynet representando o aprendizado para gerar animações [18].

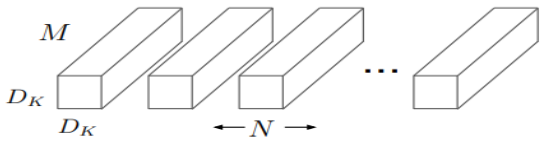
Para o processo de classificação, utilizou-se dois classificadores para cada dataset. A

Mobilenet foi a rede neural comum entre ambos os dataset, com as configurações citadas na subseção subsequente. Para o CK, utilizou-se uma CNN proposta, já que, necessitava-se de uma quantidade menor de parâmetros para o que pudesse convergir os modelos, já que a quantidade de dados disponível para o Dataset CK é muito pequena e mesmo com transferência de aprendizado, os modelos não se comportaram bem. E para o MUG, foi utilizada a Resnet50 [3], uma variação da Resnet [3] com uma quantidade menor de camadas.

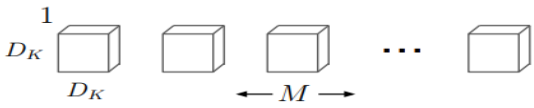
5.2 MobileNet

A MobileNet é uma poderosa rede neural, lançada em 2017, que tem tamanho reduzido para aplicações mobile e um grande poder em obter resultados precisos.

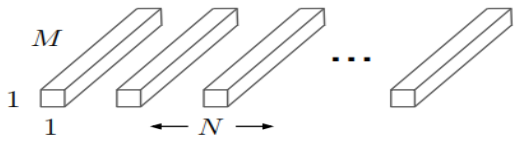
Essa arquitetura dá-se por meio da convolução em profundidade (depthwise convolution [1]) que é dada por meio de um único filtro a cada e então a convolução pontual (pointwise convolution [1]) é aplicada para combinar as saídas da convolução em profundidade. Uma convolução normal combina os dados de entrada em um novo conjunto de dados de saída em um único passo. Essa separação, tem o intuito de diminuir o tamanho do modelo, e assim, por exemplo, ser utilizada em dispositivos móveis.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figura 9: Esquema MobileNet [1].

A MobileNet [1] foi utilizada para a o processo de classificação para ambos os datasets. Para o CK [6], um dataset, que contém uma pequena quantidade de dados (cerca de 5200 imagens, para 7 classes especificadas neste projeto), a MobileNet [1] neste caso teve um congelamento de todas suas camadas, exceto para as camadas de Batch Normalization [12]. O congelamento completo tem por objetivo reduzir a quantidade de liberdade da rede durante o treinamento (reduzir a quantidade de parâmetros, visto que o dataset é muito pequeno). As próximas camadas da rede, ficaram:

CamadasPrincipais	Regularização	BatchNormalization
GlobalAveragePooling2D	Dropout(0.4)	Momentum(0.95)
Dense(7)	LL2(0.0001)	Não

Tabela 1: Configuração MobileNet para o DatasetCK [6].

Já para o dataset MUG [7], há uma quantidade maior de dados para as classes selecionadas, assim, utilizou-se uma configuração da MobileNet [1] com uma maior quantidade de parâmetros treináveis, para que assim, o modelo pudesse ser mais complexo e consequentemente produzir um resultado melhor. As configurações, ficaram:

CamadasPrincipais	Regularização	BatchNormalization
GlobalAveragePooling2D	Dropout(0.4)	Momentum(0.99)
Dense(32)	LL2(0.01)	Momentum(0.99)
Dense(6)	LL2(0.01)	Não

Tabela 2: Configuração MobileNet para o Dataset MUG [7].

5.3 Rede Neural Convolucional

Para a classificação do dataset CK [6] visto a pequena quantidade de dados disponíveis por parte deste, fez-se com que cria-se uma arquitetura de uma rede neural convolucional para a classificação do mesmo. Assim, a estrutura ficou composta por 5 camadas, sendo a primeira camada inicial de entrada, uma camada convolucional em duas dimensões com 8 filtros 3x3.

Já as três camadas subsequentes são camadas convolucionais em duas dimensões. A segunda é composta por 16 filtros 3x3. Já as próximas duas são camadas com convoluções com kernels de tamanho 1x1, com 64 e 24 respectivamente. Por fim, há camada final que contém 7 neurônios de saída (número de classes). Segue a configuração completa:

CamadasPrincipais	Regularização	BatchNormalization	Ativação	Kernels
Conv2D	Não	Não	relu	8 (3x3)
Conv2D	LL2	Não	relu	16 (3x3)
Conv2D	Dropout e LL2	Sim	relu	64 (1x1)
Conv2D	Dropout e LL2	Sim	relu	32 (1x1)
Dense	Dropout e LL2	Sim	softmax	Não

Tabela 3: Configuração Rede Neural Convolucional para o Dataset CK [7].

5.4 Resnet

A Resnet (Residual Networks) é uma rede neural introduzido em 2015 para o desafio de classificação de ImageNet. Resnet's são redes neurais extremamente profundas e produzem

grandes resultados e métricas.

Para lidar com a grande profundidade, já que este é um problema para a classificação, pois sua complexidade influi em uma dificuldade de convergência. Para resolver esse problema, introduziu-se o conceito de “skip”, que é pular algumas camadas de convolução, ou seja, ligar a saída de uma camada a uma outra camada seguinte que não seja a subsequente. Esses pulos ajudam ao gradiente não desaparecer (um dos problemas de redes neurais muito profundas) além de garantir que uma camada superior possa aprender tão bem quanto uma camada inferior.

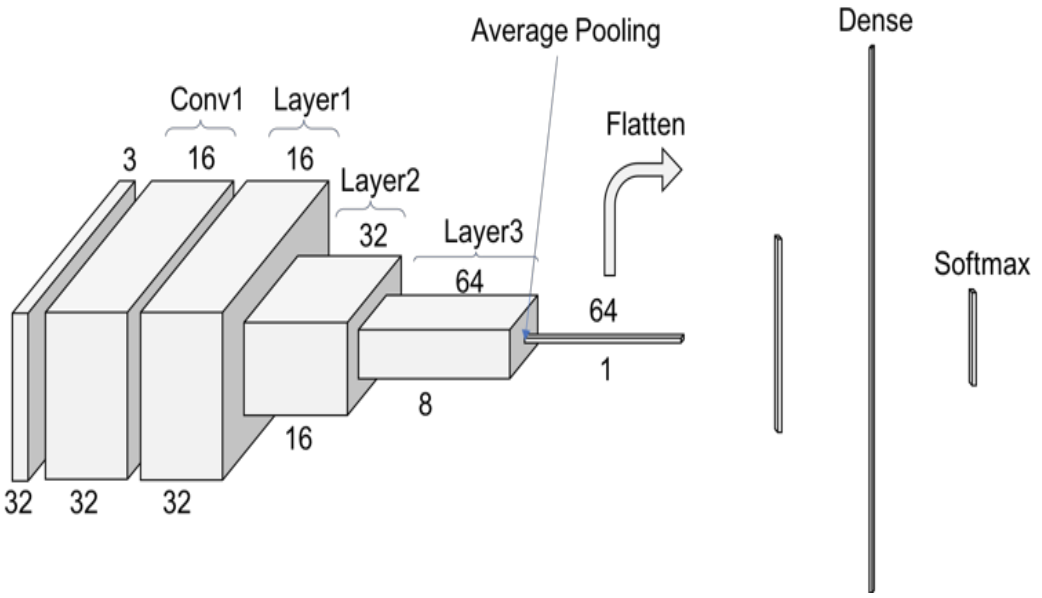


Figura 10: Modulo Resnet.

Assim, no processo de classificação, utilizou-se uma Resnet menor (50 camadas) em relação a vencedora da disputa da ImageNet em 2015, com 152 camadas.

Para a Resnet50 [3], foi avaliada somente para o dataset MUG, para isso, houve o congelamento da rede em 95 % da rede e houve a adição das seguintes camadas após isso, para completar o ajuste fino:

CamadasPrincipais	Regularização	BatchNormalization
GlobalAveragePooling2D	Não	Momentum(0.95)
Dense(6)	LL2(0.01)	Não

Tabela 4: Configuração MobileNet para o Dataset MUG [7].

6 Resultados

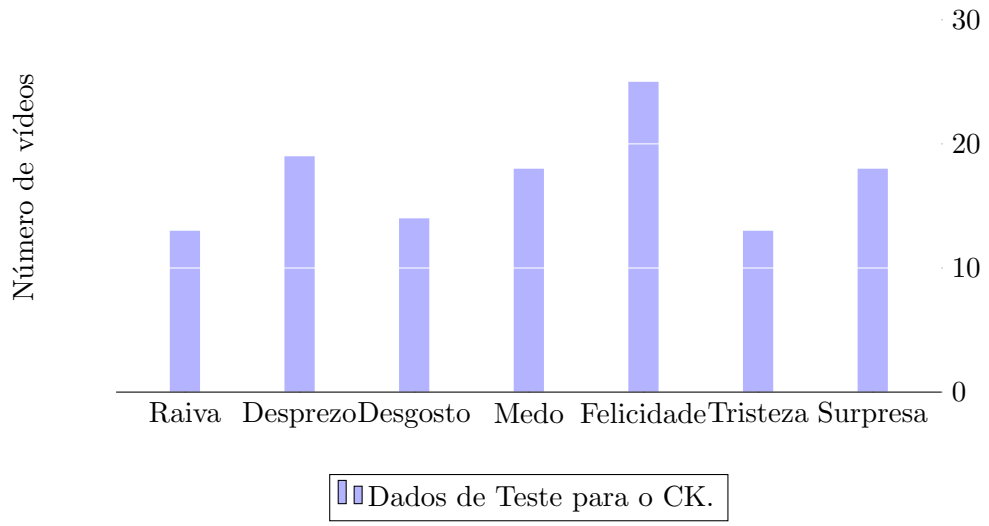
Nos casos das redes prontas, foi utilizada a transferência de aprendizado com o ajuste fino, a partir de um modelo pré-treinado em uma certa base de dados e então, pode-se utilizar desses fatores para não ter que refazer o treinamento todo da rede, que em redes complexas, pode gerar casos de sobre-treinamento vista sua complexidade em relação ao tamanho da base de dados. Assim, utilizou-se cada umas das três respectivas redes neurais MobileNet [1] e Resnet [3] treinadas na base de dados do ImageNet.

Para todo o processo de classificação, foram utilizados os parâmetros apresentados na seção anterior. Além disso, para cada processo, utilizou-se de 50 épocas (número de vezes em que o dataset completo é passado pela rede neural) e o batch foi de 32 (aproximando o número de frames por vídeo), como intuito de que em uma iteração a rede veja todo o vídeo).

6.1 Resultados para CK

Para o dataset do CK [6], utilizou-se dois modelos para classificação. A MobileNet [1], utilizando a transferência de aprendizado juntamente com o ajuste fino e uma rede neural convolucional proposta, com o intuito de analisar os resultados obtidos. A distribuição abaixo, com 120 vídeos, mostra os dados utilizados no processo de teste dos modelos para o dataset:

Gráfico 1: Distribuição de Dados do dataset CK.



Para a Mobilenet, os resultados obtidos foram:

-	Raiva	Desprezo	Desgosto	Medo	Felicidade	Tristeza	Surpresa
Raiva	0,38	0,00	0,15	0,15	0,23	0,00	0,08
Desprezo	0,21	0,37	0,16	0,05	0,11	0,05	0,05
Desgosto	0,14	0,00	0,43	0,07	0,07	0,07	0,21
Medo	0,22	0,00	0,11	0,39	0,06	0,17	0,06
Felicidade	0,12	0,04	0,12	0,08	0,32	0,08	0,24
Tristeza	0,38	0,00	0,15	0,00	0,00	0,31	0,15
Surpresa	0,22	0,11	0,06	0,11	0,17	0,06	0,28

Tabela 5: Tabela de Confusão para MobileNet e CK sem dados sintéticos normalizada em decimais.

Com a adição de dados sintéticos no processo de treinamento, o resultado obtido foi de:

-	Raiva	Desprezo	Desgosto	Medo	Felicidade	Tristeza	Surpresa
Raiva	0,46	0,08	0,15	0,15	0,08	0,00	0,08
Desprezo	0,11	0,37	0,11	0,16	0,11	0,16	0,00
Desgosto	0,14	0,14	0,43	0,07	0,00	0,14	0,07
Medo	0,06	0,22	0,11	0,33	0,11	0,11	0,06
Felicidade	0,04	0,00	0,12	0,16	0,36	0,12	0,20
Tristeza	0,08	0,23	0,00	0,08	0,00	0,54	0,08
Surpresa	0,06	0,06	0,06	0,11	0,17	0,06	0,44

Tabela 6: Tabela de Confusão para MobileNet e CK sem dados sintéticos normalizada em decimais.

Assim, a acurácia normalizada obtida para a MobileNet [1] sem a adição de dados sintéticos foi de: 35,42 % E quando adicionou-se foi de: 41,50 %. Já, quando utilizou-se a CNN, as matrizes de confusão foram:

-	Raiva	Desprezo	Desgosto	Medo	Felicidade	Tristeza	Surpresa
Raiva	0,38	0,23	0,15	0,08	0,08	0,00	0,08
Desprezo	0,16	0,37	0,11	0,11	0,16	0,11	0,00
Desgosto	0,07	0,29	0,43	0,00	0,07	0,07	0,07
Medo	0,06	0,33	0,11	0,33	0,17	0,00	0,00
Felicidade	0,08	0,20	0,12	0,08	0,28	0,08	0,16
Tristeza	0,00	0,38	0,15	0,08	0,00	0,38	0,00
Surpresa	0,11	0,22	0,11	0,17	0,06	0,11	0,22

Tabela 7: Tabela de Confusão para MobileNet e CK sem dados sintéticos normalizada em decimais.

Com a adição de dados sintéticos no processo de treinamento, o resultado obtido foi de:

-	Raiva	Desprezo	Desgosto	Medo	Felicidade	Tristeza	Surpresa
Raiva	0,43	0,14	0,07	0,07	0,14	0,07	0,07
Desprezo	0,06	0,33	0,17	0,06	0,22	0,17	0,00
Desgosto	0,07	0,07	0,57	0,00	0,14	0,00	0,07
Medo	0,11	0,17	0,11	0,28	0,11	0,11	0,11
Felicidade	0,04	0,04	0,24	0,04	0,36	0,08	0,20
Tristeza	0,00	0,23	0,08	0,00	0,15	0,38	0,15
Surpresa	0,11	0,00	0,17	0,06	0,22	0,11	0,33

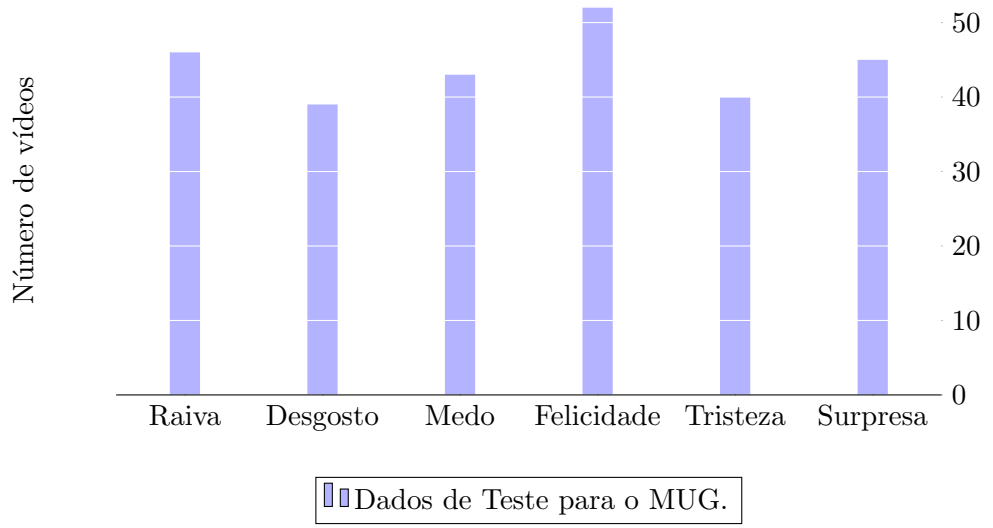
Tabela 8: Tabela de Confusão para MobileNet e CK sem dados sintéticos normalizada em decimais.

Assim, a acurácia normalizada obtida para a CNN sem a adição de dados sintéticos foi de: 36,32 % Já na adição foi de: 39,26 %

6.2 Resultados para MUG

Primeiramente para o dataset MUG [7], executou-se os os classificadores Mobilenet [1] e Resnet50 [3]. Em ambos os casos, testou-se com e sem dados sintéticos. Para os testes, considerou-se 265 vídeos, de acordo com a distribuição:

Gráfico 4: Distribuição de Dados de Teste para o MUG.



Assim, após o teste para a MobileNet [1], obteve-se as matrizes de confusão:

-	Supresa	Tristeza	Medo	Raiva	Desgosto	Felicidade
Surpresa	0,65	0,07	0,02	0,09	0,09	0,09
Tristeza	0,13	0,45	0,08	0,13	0,08	0,15
Medo	0,05	0,07	0,47	0,16	0,05	0,21
Raiva	0,00	0,09	0,07	0,67	0,07	0,11
Desgosto	0,00	0,05	0,03	0,10	0,77	0,05
Felicidade	0,02	0,08	0,06	0,17	0,00	0,67

Tabela 9: Tabela de Confusão para MobileNet e MUG sem dados sintéticos normalizada em decimais.

-	Supresa	Tristeza	Medo	Raiva	Desgosto	Felicidade
Surpresa	0,67	0,09	0,09	0,00	0,09	0,07
Tristeza	0,18	0,53	0,03	0,08	0,08	0,11
Medo	0,05	0,02	0,56	0,19	0,07	0,12
Raiva	0,13	0,04	0,04	0,67	0,02	0,09
Desgosto	0,00	0,05	0,05	0,05	0,74	0,10
Felicidade	0,06	0,10	0,06	0,02	0,08	0,69

Tabela 10: Tabela de Confusão para MobileNet e MUG com dados sintéticos normalizada em decimais.

Obtendo assim uma acurácia normalizada de: 64,34 % com a adição de dados sintéticos e uma acurácia de 61,34 % para o dataset original.
Já para o modelo da Resnet, obteve-se as matrizes de confusão:

-	Supresa	Tristeza	Medo	Raiva	Desgosto	Felicidade
Surpresa	0,64	0,07	0,02	0,07	0,11	0,09
Tristeza	0,13	0,50	0,08	0,13	0,08	0,10
Medo	0,09	0,07	0,42	0,23	0,5	0,14
Raiva	0,4	0,04	0,09	0,65	0,09	0,09
Desgosto	0,00	0,03	0,03	0,15	0,69	0,10
Felicidade	0,06	0,12	0,08	0,13	0,02	0,60

Tabela 11: Tabela de Confusão para Resnet e MUG sem dados sintéticos normalizada em decimais.

-	Supresa	Tristeza	Medo	Raiva	Desgosto	Felicidade
Surpresa	0,69	0,07	0,09	0,02	0,07	0,07
Tristeza	0,08	0,55	0,15	0,08	0,05	0,10
Medo	0,05	0,02	0,51	0,19	0,09	0,14
Raiva	0,11	0,04	0,07	0,67	0,04	0,07
Desgosto	0,03	0,05	0,08	0,03	0,74	0,08
Felicidade	0,08	0,15	0,10	0,08	0,04	0,56

Tabela 12: Tabela de Confusão para Resnet e MUG com dados sintéticos normalizada em decimais.

Com um valor de 58,41 % para o dataset original, e quando adiciona-se dados sintéticos, o resultado obtido passa a ser de: 62,5 %.

7 Análise e Conclusão

Podemos observar que, ao adicionarmos dados sintéticos em todos os casos a acurácia do teste sempre aumenta, já que, os datasets, principalmente o CK [6] não é balanceado, e algumas classes, como Desprezo têm significativamente menor amostras que as demais em relação ao dataset original.

-	CK sem dados sintéticos	CK com dados sintéticos
MobileNet	35,42 %	41,50 %
CNN Proposta	36,32 %	39,26 %
-	MUG sem dados sintéticos	MUG com dados sintéticos
MobileNet	61,34 %	64,34 %
Resnet50	58,41 %	62,50 %

Tabela 13: Resultados da classificação com e sem dados sintéticos

Primeiramente, para o dataset CK [6] houve a necessidade de trabalhar com uma quantidade de parâmetros menor que para os demais datasets, já que, a quantidade de dados disponível era consideravelmente baixa. Dado isso, podemos observar que a acurácia não ficou alta, dado a dificuldade de aprender sobre os dados. Porém, quando adiciona-se os dados sintéticos, para a MobileNet [1], por exemplo, podemos observar que há um ganho de 6 %, já que há um aumento considerável no número de dados. O mesmo já não ocorre na CNN proposta, com um ganho de 3 %. Isso pode ser explicado que, um modelo como MobileNet, utilizado no processo de transferência de aprendizado com ajuste fino, pode oferecer melhores resultados e condições para o processo de classificação, tendo os parâmetros melhores adaptados ao problema.

Já, quando lida-se com os dados do dataset MUG [7] onde há uma maior disponibilidade

de dados, a acurácia é maior, já que o modelo pode-se adaptar melhor à rede. Assim, para a MobileNet [1], pode-se observar que houve um ganho de 3 % com os dados sintéticos. Uma vez que, a MobiletNet não é uma rede neural muito grande, então o aumento de dados não pode ser significativamente melhor e também, o dataset já está um pouco balanceado em relação ao CK [6] por exemplo. Quando analisa-se a Resnet, mesmo a acurácia sendo menor, há um ganho de cerca de 4 %, também não é muito grande, porém mostra que em um classificador mais complexo com um maior grau de liberdade de aprendizado, uma maior quantidade de dados pode ajudar sim no processo classificatório.

Assim, podemos concluir que, ao adicionar dados sintéticos para balancear datasets, os resultados podem sim melhorar os classificadores, visto que, a falta de balanceamento de datasets é um grande desafio no processo de classificação, que pode piorar o desempenho de um modelo. E também, podemos concluir que os dados criados tem uma ótima qualidade, já que influenciaram diretamente no resultado da classificação.

Referências

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

[2] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna
Rethinking the Inception Architecture for Computer Vision

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
Deep Residual Learning for Image Recognition

[4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton
ImageNet Classification with Deep Convolutional Neural Networks

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
Generative Adversarial Networks

[6] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar
The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression

[7] N. Aifanti, Christos Papachristou, Anastasios Delopoulos
The MUG facial expression database

[8] Saad Albawi ; Tareq Abed Mohammed ; Saad Al-Zawi
Understanding of a convolutional neural network

[9] Herbert Robbins, Sutton Monroi
A Stochastic Approximation Method

[10] Corinna Cortes, Mehryar Mohri, Afshin Rostamizadeh
L2 Regularization for Learning Kernels

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov
Dropout: A Simple Way to Prevent Neural Networks from Overfitting

[12] Sergey Ioffe, Christian Szegedy
Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

[13] Antreas Antoniou, Harrison Edwards and Amos Storkey *DATA AUGMENTATION GENERATIVE ADVERSARIAL NETWORKS*

[14] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan *GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification*

[15] Haşim Sak, Andrew Senior, Françoise Beaufays *Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition*

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei *Large-Scale Video Classification with Convolutional Neural Networks, In Proc. IEEE*

[17] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley *Least Squares Generative Adversarial Networks, In Proc. IEEE*

[18] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe *Animating Arbitrary Objects via Deep Motion Transfer, In Proc. IEEE*