Inspeção Visual Automatizada de Equipamentos na Indústria para Detecção de Anomalias

André Eiki Hiratsuka
UNIFESP – Universidade Federal de São Paulo
São José dos Campos, Brasil
andre.hiratsuka@unifesp.br

Thiago Roberto Fernandes Rocha
UNIFESP – Universidade Federal de São Paulo
São José dos Campos, Brasil
thiago.roberto04@unifesp.br

***Abstract* — A Inspeção Visual Automatizada (IVA) utiliza processamento de imagens para detectar defeitos e garantir a qualidade dos produtos. A automação dessa inspeção evita problemas e erros humanos, além de melhorar o desempenho e reduzir os custos de fabricação. Com o avanço das inteligências artificiais e reconhecimento de padrões, é possível criar um sistema de inspeção automatizada em tempo real, consistente e confiável.**

Palavras-Chave — imagens, indústria, inteligência artificial, inspeção visual.

I. Introdução

Em qualquer produto, é necessário que ele atenda a um determinado padrão de qualidade para que não ocorra imprevistos. Dessa forma, visando completar uma etapa fundamental no processo de fabricação e inspeção do produto, a Inspeção Visual Automatizada (IVA) é um método de processamento de imagem para o controle de qualidade de um produto, uma técnica que visa a detecção óptica de defeitos, para a garantia de que o produto analisado esteja funcionando de acordo com as especificações.

Esse trabalho de inspeção, se realizado por seres humanos, pode apresentar diversos empecilhos e erros, pois este é um trabalho demorado e desgastante, além da necessidade de treinamento e capacitação dos trabalhadores. Ademais, o rendimento dos indivíduos pode ser afetado por fadiga ou problemas pessoais, tanto pela carga pesada de trabalho quanto por eventos inesperados. Por causa disso, a ideia de automatizar essa inspeção torna-se muito interessante, tendo em vista a prevenção de possíveis problemas de inspetores humanos, além da melhora do rendimento do produto e redução do custo de fabricação.

Com o crescente desenvolvimento das inteligências artificiais, juntamente com o reconhecimento de padrões e processamento de imagens, torna-se palpável e viável a elaboração e criação de um sistema de inspeção automatizada capaz de operar em tempo real de forma consistente, robusta e confiável.

II. Motivação

Com a inovação tecnológica da indústria 4.0, é possível extrair conhecimento a partir de estratégias inteligentes e adoção de monitoramento e fusão de dados, além da aplicação de métodos de aprendizado de máquina e otimização. Do ponto de vista da ciência de dados, a motivação é prever anomalias em máquinas, equipamentos e processos de forma eficiente, com a finalidade de antecipar erros e evitar danos do produto defeituoso, como realçado por Lu [1].

III. Conceitos Fundamentais

A. *Aprendizado de Máquina*

Segundo Mitchell [2], o aprendizado de máquina é uma subárea da inteligência artificial, que utiliza técnicas computacionais no desenvolvimento de sistemas capazes de tomar decisões a partir do conhecimento adquirido pelos dados fornecidos e obtidos nas experiências passadas. O aprendizado pode ser dividido em: aprendizado supervisionado e não supervisionado.

No aprendizado supervisionado, o algoritmo recebe um conjunto de dados chamado de conjunto de treinamento, com o objetivo de predizer um rótulo já conhecido, dado um conjunto de exemplos esse algoritmo criará um estimador, como declarado por Gama et al [3]. Nesse aprendizado, as duas principais tarefas são a de classificação e de regressão. Na classificação, lida-se com valores discretos ou categóricos, enquanto na regressão os valores são contínuos.

Já no aprendizado não supervisionado, não há uma saída ou uma verdade conhecida para predizer. Nesse tipo de aprendizado, estão contidas principalmente atividades de agrupamento, como o K-Means e o K-Medoids.

B. *Redes Neurais Profundas*

As redes neurais artificiais usam como referência o funcionamento do cérebro humano, sendo a unidade mais básica dele o neurônio. Um neurônio possui o dendrito, o axônio e o corpo celular. De acordo com de Padua Braga [4], as Redes Neurais Artificiais são projetadas com o objetivo de modelar a maneira como o cérebro realiza uma

tarefa ou função e com isso resolver problemas de diversas categorias como classificação, agrupamento ou regressão.

Um Multilayer Perceptron (MLP) é composto por 3 camadas: a entrada de dados, a camada oculta e a camada de saída, sendo que a quantidade de camadas ocultas pode variar. Cada camada é composta por um "neurônio", que se liga desde a entrada até a saída a partir das "sinapses", como visto por Hassaballah e Awad [5].

Algumas redes existentes são:

A rede ResNet: tendo em vista que redes neurais mais profundas podem apresentar problemas de degradação, a ResNet, do inglês Residual Network, é uma arquitetura, cuja técnica consiste em fazer com que a saída de uma camada seja usada como entrada da próxima camada convolucional e também seja usada como entrada em uma camada mais profunda, como se a informação pulasse algumas camadas para ser processada por camadas mais profundas [19], como pode ser observado na Figura 2.

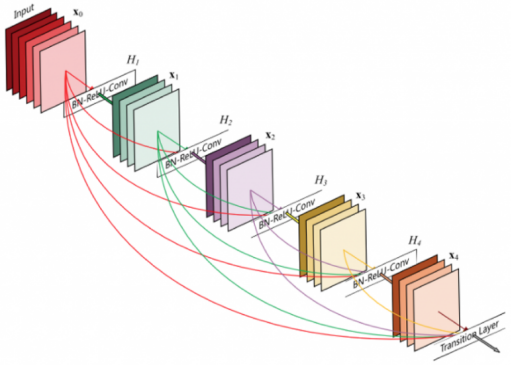

Fig 1. Arquitetura da rede ResNet [19].

A partir deste conceito, camadas adicionadas se tornam o mapeamento de identidade e as outras camadas são copiadas do modelo mais raso aprendido. Esta solução indica que um modelo mais profundo não deve produzir um erro de treinamento maior do que sua versão da rede mais rasa.

A rede NASnet: a Neural Architecture Search Network (NASnet), e uma arquitetura cuja finalidade é a de construir um bloco com alta performance para a classificação de pequenos conjuntos de imagens e então generalizar este bloco para um conjunto maior. Desta forma, atingindo uma alta capacidade de categorização para uma quantidade reduzida de parâmetros [20].

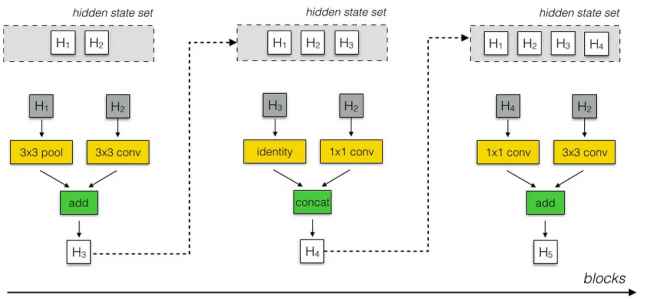

Fig 2. Pesquisa espacial de uma rede NASnet [20].

Esta rede é composta por blocos normais e blocos de redução, sendo que o último diferencia-se por reduzir a resolução de suas camadas (veja Figura 4). Uma estrutura chamada Recurrent Neural Network (RNN) pesquisa pela estrutura destes blocos e prediz recursivamente o restante desta estrutura a partir de dois estados ocultos fornecidos previamente. A RNN procura por diferentes operações para combinar estes dois estados anteriores e então gerar novos estados ocultos com base nas camadas anteriores.

A rede Vision Transformer (ViT) surgiu recentemente como uma rede neural convolucional que demonstrou resultados excelentes se comparada com outras redes mais usuais. Quando pré-treinada em quantidades altas de dados e transferida para imagens de tamanho médio/pequeno, a ViT conseguiu bons resultados com relativamente menos uso de recursos computacionais [21]. A ViT é uma CNN que utiliza os chamados transformadores, ou transformers, para o processamento de imagens.

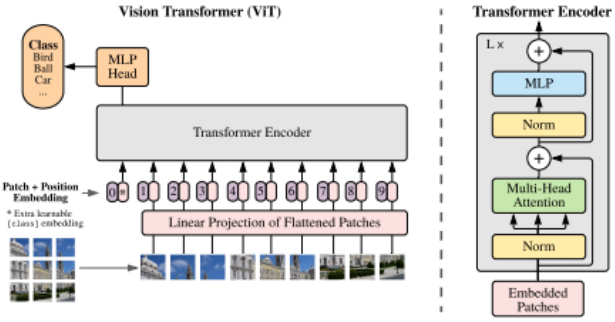

Fig 3. Visão geral do modelo [21]

Na ViT, divide-se a imagens em patches de tamanho igual, chamados patches. Esses patches servirão como a entrada do modelo. Em seguida, eles serão alocados linearmente em vetores de características e direcionados para o Transformer Encoder. O encoder é um bloco repetido de camadas de atenção multihead (multi-head self-attention) e redes neurais totalmente conectadas (feed-forward). As multihead permitem que os patches se comuniquem uns com os outros, capturando as relações de contexto em toda a imagem. Assim, esse processamento é repetido várias vezes para melhorar a representação dos patches.

A rede VGG: é uma arquitetura para redes neurais convolucionais, desenvolvida pelo Visual Geometry Group da Universidade de Oxford. Esta rede utiliza filtros de tamanho 3 × 3 para suas camadas de convolução, empilhadas umas sobre as outras.

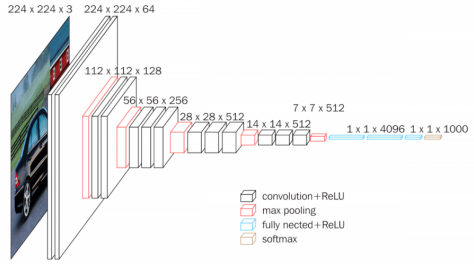

Fig 4. Arquitetura da rede VGG [23].

Nota-se na Figura 1, que ao término de uma pilha de camadas de convolução é realizada uma camada de pooling, para reduzir as dimensões dos filtros. Ao término das camadas de convolução e pooling, é chamada uma camada de flattening para transformar a matriz em um vetor e, por fim, é chamada uma função softmax para verificar a probabilidade daquele determinado ponto da imagem pertencer a uma determinada classe [23].

O MobileNet V2 surgiu como uma versão aprimorada da rede MobileNet. Apesar de os dois continuarem com vantagem da capacidade de execução mesmo com recursos computacionais limitados, a versão V2 contém alguns aprimoramentos.

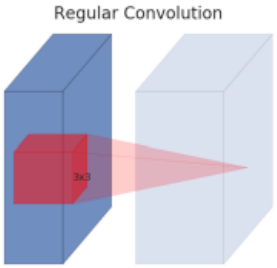

Fig 5. Convolução regular do MobileNet V2 [24]

Assim como podemos ver na figura, o Mobile Net utiliza blocos de convolução, ou bottlenecks. Ela possui 3 camadas: um bottleneck 1 x 1 para reduzir o tamanho, uma "depthwise separable convolution", e novamente um bloco 1 x 1 para voltar ao tamanho original.

Em relação ao depthwise separable convolution, ao invés da convolução tradicional que envolve todos os canais de entrada em cada pixel, a MobileNetV2 possui duas etapas: Primeiro, aplica-se um kernel separado a cada canal de entrada individualmente. Em seguida, uma convolução ponto a ponto (pointwise) é aplicada para combinar as saídas da depthwise em um novo espaço de características [24].

A rede Inception: tem como características o uso de filtros para convoluções 1×1, 3 × 3 e 5 × 5 dentro de um mesmo nível da rede, acarretando em mais camadas paralelas e menos camadas profundas, outro fator é a adição de filtros 1 × 1 em todas as camadas convolucionais com o intuito de reduzir a dimensionalidade da rede e aumentar o desempenho.

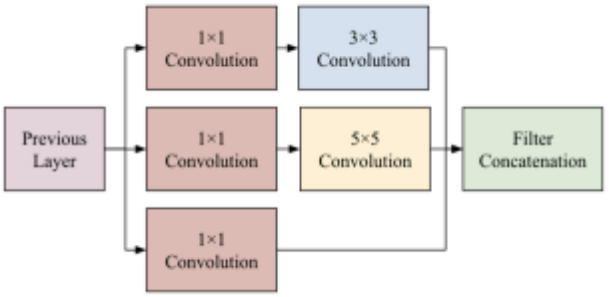

Fig 6. Arquitetura da rede Inception [25].

Esta arquitetura, exibida na Figura 3, também caracteriza-se pelo uso do Average-Pooling ao invés de camadas totalmente conectadas na parte superior da CNN, o que elimina uma grande quantidade de parâmetros menos importantes [Silva 2018]. Por apresentar menos parâmetros, este tipo de arquitetura necessita de menos recursos computacionais enquanto apresenta um bom desempenho.

A rede EfficientNet: ao contrário de outras arquiteturas que decidem de maneira arbitrária o método ao qual vão escalonar seu número de camadas, a EfficientNet, utiliza do método de escalonamento por composição, que consiste em alternar entre escalonamento por largura, profundidade e resolução, haja visto que todos estes métodos começam a perder eficiência a partir de um determinado número de camadas. Um exemplo desta arquitetura é mostrado na Figura 5.

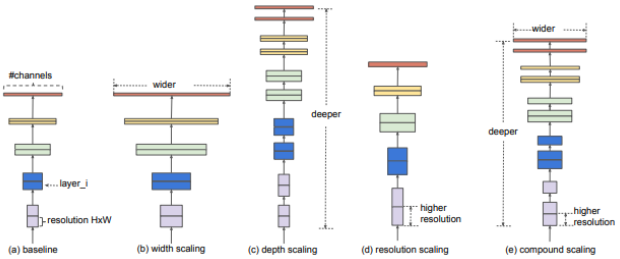

Fig 7. Modelos de escalonamento [26].

O método de composição baseia-se em uma razão constante, que depende de três parâmetros, sendo eles alfa, beta e gama, ou profundidade, largura e resolução. Todos estes parâmetros são exponenciados por um parâmetro phi.

C. Redes Neurais Convolucionais

As Convolutional Neural Networks (CNNs) são redes que aplicam a operação de convolução em uma de suas

camadas. Esse tipo de rede tem sido bastante utilizado em problemas onde as predições são realizadas com base em dados que possuem topologia de grade e dependência espacial como séries temporais (1D) e imagens (2D). Segundo Goodfellow [6], a operação de convolução consiste na aplicação de um filtro sobre os dados de entrada, sendo basicamente uma operação linear que multiplica a entrada por um conjunto de pesos.

Comumente após a aplicação das camadas de convolução, são utilizadas funções de ativação, que tem como objetivo aplicar uma não linearidade nas camadas geradas a partir das convoluções. A não linearidade das camadas intermediárias permite que as aplicações sucessivas dessas distorções tornem as categorias de saída linearmente separáveis, como declarado por Vargas et al [7]. Além das funções de ativação, há as funções de pooling (agrupamento), cujo objetivo é reduzir a dimensionalidade dos dados, permitindo com que a eficiência computacional aumente enquanto as características do dado não se percam. O uso das camadas de pooling auxilia em uma melhor generalização, o que gera uma maior confiabilidade para o algoritmo ao lidar com dados desconhecidos. Outra função importante é a função de flattening, que é utilizada para transformar uma matriz recebida de camadas anteriores em um vetor.

D. Métodos, técnicas, abordagens e estratégias

Matriz de confusão ou tabela de contingência é uma técnica que compara os resultados de classificação obtidos na predição comparado aos valores reais. Na predição, existem 4 tipos de erro: ao fazer a predição, prevê-se verdadeiro ou falso e o resultado pode ser tanto verdade quanto falso. Dessa forma, a tabela ficará neste formato geral:

	Real Positivo	Real Negativo
Predito Positivo	True Positive (TP)	False Positive (FP)
Predito Negativo	False Negative (FN)	True Negative (TN)

Fig 8. Exemplo de uma matriz de confusão.

Uma métrica bastante utilizada utilizando os conceitos da matriz de confusão é o F1 Score. Ela é dada pela seguinte fórmula:

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

Ela utiliza a junção de outros dois conceitos: O precision e o recall, conseguindo oferecer uma visão mais geral do desempenho do modelo. As fórmulas de precisione recall são dadas por:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Pelas fórmulas, é possível perceber o objetivo de cada uma. A precision deseja analisar o total de acertos em relação aos preditos positivos. Já o recall analisa o total de acertos em relação ao número de dados realmente positivos. Uma taxa alta de precision e recall geralmente indicam uma baixa taxa de falsos positivos e falsos negativos respectivamente.

Para ajustar os pesos durante um treinamento, é possível utilizar otimizadores. Foi utilizado o otimizador Adam e o RMSprop. O otimizador Adam é uma técnica muito eficiente quando envolve muitos dados ou parâmetros. Ele mistura duas metodologias de descida pela gradiente.

Momentum: Usando a média exponencialmente ponderada, faz com que o algoritmo convirja para os mínimos de forma mais rápida

$$w_{t+1} = w_t - \alpha m_t$$, onde

$$m_t = \beta m_{t-1} + (1 - \beta)\left[\frac{\delta L}{\delta w_t}\right]$$.

Root mean square prop ou RMSprop: utiliza a média móvel exponencial.

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t + \varepsilon)^{1/2}} * \left[\frac{\delta L}{\delta w_t}\right]$$, onde

$$v_t = \beta v_{t-1} + (1 - \beta) * \left[\frac{\delta L}{\delta w_t}\right]^2$$

Assim, obtemos a fórmula geral do Adam:

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}} + \varepsilon}\right)$$

Função de ativação, de modo geral, é uma função matemática que é aplicada à entrada do neurônio artificial. Utilizamos a função de ativação softmax em nossa rede neural. A fórmula é dada por:

$$\phi_i = \frac{e^{z_i}}{\sum_{j \in group} e^{z_j}}$$

Em um neurônio, ao invés de classificar a partir de um valor inteiro, a função atribui uma probabilidade de pertencer a cada classe.

Em relação ao erro do treinamento do modelo, podemos utilizar funções de perda para calcular o erro ou perda dele. Em nosso caso, utilizamos o Loss Categorical_Crossentropy.

Loss Categorical_Crossentropy ou perda de entropia cruzada categórica é uma função de perda que compara a probabilidade prevista na rede neural com a probabilidade correta do treino, indicando o quanto o treino está se ajustando aos dados. A fórmula é dada por:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\ \text{size}}} y_i \cdot \log \hat{y}_i$$

Ao treinarmos uma rede, a rede pode se ajustar excessivamente ao conjunto de treinamento, causando overfitting ou sobreajuste. Quando ocorre isso, o conjunto está tão acostumado com o treino, ficando muito específico ao invés de generalizar, que terá uma queda de acurácia no conjunto de teste.

Para reduzir o overfitting, utilizamos duas técnicas: camadas de dropout e early stopping. Na camada de dropout, desativam-se aleatoriamente conjuntos de neurônios no treinamento. Dessa forma, consegue-se uma melhor generalização do conjunto de treinamento, visto que não se deve depender individualmente desses neurônios desativados para encontrar padrões.

No early stopping, há uma análise do treinamento do conjunto. Ao monitorar o modelo, quando o desempenho começa a cair e o modelo começa a se ajustar demais, interrompe-se o treino. Desse modo, pode-se ter critérios pré-determinados de parada, como o aumento do erro ou diminuição da acurácia.

Além disso, para diminuir a complexidade e os parâmetros da rede neural, utilizamos o average global layer. Ele calcula a média global dos recursos em um tensor tridimensional, transformando em um vetor unidimensional, diminuindo os parâmetros e, consequentemente, reduzindo o overfitting.

Caso a base de dados esteja muito desbalanceada, ou seja, algumas categorias de dados estejam mais predominantes do que outras, pode-se aplicar a técnica de undersampling para tentar resolver o problema. É possível, por exemplo, pegar um número igual de todas as classes de um conjunto de dados, ou então selecionar um grande número de dados apenas das classes mais populosas.

Após um pré-treinamento, é possível ajustar o modelo a um novo conjunto de dados para outra tarefa em específico. O nome dessa ação é fine tuning, e essa técnica é muito útil para uma maior generalização do treino, além de utilizar menos recursos computacionais, pois o conjunto já é pré-treinado.

IV. Trabalhos Relacionados

- O livro “Machine Learning” de Tom M. Mitchell [2] aborda sobre o conceito de aprendizagem de máquina. Para realizar este projeto de inteligência artificial, é extremamente necessário que haja uma base boa de conhecimento, principalmente de aprendizado de máquina e redes neurais profundas;

- Em segundo plano, os autores Goodfellow e Bengio [6] abordam detalhadamente sobre redes neurais e Deep Learning, outro conceito imprescindível para o entendimento e realização das IVA. Este livro é outro exemplo de obra em que não está relacionado diretamente com processamento de imagem e detecção de anomalias, mas os conceitos, como redes convolucionais e aprendizado profundo, complementam de forma rica o trabalho;

- Em relação a trabalhos parecidos com a nossa proposta, na obra “Deep Residual Learning for Image Recognition” [8] publicada pela própria IEEE, pelos autores He, Zhang, Ren e Sun, foram utilizadas redes neurais, conceito desejado neste projeto de IVA, para classificação de imagens. Apesar do uso de redes neurais não ser uma tarefa fácil, a existência de outros trabalhos parecidos é bastante útil para utilizar como referência;

- Os autores Huang e Pan colocam em prática a teoria de IVA em semicondutores, no artigo “Automated visual inspection in the semiconductor industry: A survey.” [9]. Na pesquisa, foi utilizado a captura de imagens, comparando a diferença entre um produto de boa qualidade, identificando algum possível defeito pelos padrões encontrados, dividido em 4 técnicas adotadas;

- O câncer de mama acaba sendo um problema muito grave caso não seja diagnosticado a tempo, um TCC feito na Universidade Federal de Uberlândia, chamado “Classificação de lesões em imagens histológicas de mama, usando wavelet e resnet-50” [10] busca encontrar padrões nas imagens histológicas das mamas, utilizando a rede ResNet, umas das possíveis arquiteturas de redes neurais a serem utilizadas em nosso projeto;

- Em “Um Estudo sobre Redes Neurais Convolucionais e sua Aplicação em Detecção de Pedestres” [7], os autores unem o aprendizado profundo junto ao olhar da visão computacional, e comparam técnicas de ambas as áreas, a fim de desenvolver um algoritmo capaz de detectar

regiões em imagens digitais ocupadas por pedestres;

- Na Universidade Federal do Ceará, o estudante Rodrigo Valentin escreveu um TCC sobre "Um estudo comparativo entre redes neurais convolucionais para a classificação de imagens" [11]. Neste TCC, o estudante utilizou 3 redes neurais diferentes: A ResNet, VGG e Inception. Fazendo um comparativo entre essas redes, chegou-se à conclusão de que o modelo ResNet teve um rendimento melhor do que os outros;

- Em "Bag of Tricks for Image Classification with Convolutional Neural Networks" [12], os autores exaltam a importância do refinamento em procedimentos de treinamento, onde eles combinam diversas técnicas diferentes a fim de melhorar algumas das principais e mais conhecidas arquiteturas de redes neurais profundas;

- A MobileNetV2 é uma arquitetura de rede neural convolucional desenvolvida por Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov e Liang-Chieh Chen no artigo "MobileNetV2: Inverted Residuals and Linear Bottlenecks" [13]. Esta arquitetura é acentuada por sua estrutura de bloco residual invertido, a qual utiliza de camadas mais finas de bottleneck nas entradas e saídas dos blocos residuais;

- No artigo "Review: NASNet — Neural Architecture Search Network (Image Classification)" [14], do autor Sik-Ho Tsang, há um estudo da qualidade da rede NasNet, comparando com outras redes famosas, como ResNet, Inception e MobileNet. Eles conseguiram mostrar diferentes exemplos de como o NASNet conseguiu apresentar um desempenho igual ou até mesmo melhor do que outros.

- O artigo "A review of applications of visual inspection technology based on image processing in the railway industry" [15] de Liu,Wang e Luo analisa a aplicação da IVA nas ferroviárias de diversos outros artigos feitos, e analisa o futuro desse ramo. Os autores fizeram algumas conclusões, como a eficiência alta do uso da IVA, mas também afirmando que possui seus devidos problemas, sendo necessário um investimento maior no futuro.

- Há também outros estudos de inspeções visuais em pontes, como o artigo "Drone-Enabled Bridge Inspection Methodology and Application" [16]. O uso de drones foi usado para o auxílio na coleta de dados para a inspeção. O teste foi um sucesso, e os drones conseguem capturar imagens de alta qualidade para análise, podendo ser usado em áreas de difícil acesso a humanos.

- No artigo "A cascading fuzzy logic with image processing algorithm–based defect detection for automatic visual inspection of industrial cylindrical object's surface" [17], utiliza-se a IVA em objetos cilíndricos industriais, obtendo uma acurácia de mais de 80% na detecção dos objetos defeituosos.

- A rede ViT já foi utilizada no artigo "When CNN Meet with ViT: Towards Semi-Supervised Learning for Multi-Class Medical Image Semantic Segmentation" [18], auxiliando na segmentação semântica de imagens médicas multiclasse. Os resultados demonstraram alta sensibilidade e especificidade, duas medidas de avaliação utilizadas.

V. Objetivo

O principal objetivo deste projeto é desenvolver modelos de inspeção visual em equipamentos para detectar anomalias. Mais especificamente, o projeto tem como propósito o estudo e identificação das principais técnicas de aprendizado de máquina e redes neurais profundas para aplicação em inspeção visual, pré-processar bases de dados relacionadas ao processo de monitoramento de equipamentos, realizar experimentos de classificação de imagens e avaliação dos resultados e, por fim, identificar os defeitos e anomalias dos equipamentos.

VI. Metodologia Experimental

A figura 9 refere-se a um pipeline que representa os principais pontos de nosso modelo, incluindo entrada de dados, camadas de processamento, protocolo de validação e a saída final.

Esse projeto será desenvolvido na linguagem de programação Python, pois além de ser uma linguagem com inúmeras bibliotecas que auxiliam o desenvolvedor, ela possui uma comunidade ativa.

As bibliotecas mais atraentes para o projeto serão:

- TensorFlow: uma biblioteca de código aberto para aprendizado de máquina;
- Keras: API de alto nível para construir e treinar modelos de redes neurais;
- NumPy: pacote básico para computação científica em Python;
- scikit-learn: uma biblioteca de código aberto para aprendizado de máquina.

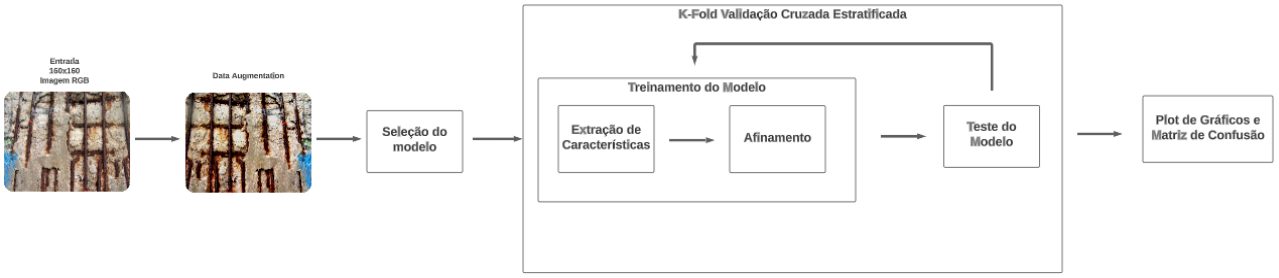

Fig 9. Pipeline descrevendo um resumo do projeto.

A. *Base de Dados*

Utilizaremos a base de dados "Bridge visual inspection dataset and approach for damage detection" [22], uma base de dados constituída por diversas imagens de danos em pontes, optamos por esta base pois ela é uma das bases que mais se aproximam do tema deste trabalho, destacando-se das demais por sua quantidade superior de dados. Além disso, os dados disponibilizados são úteis para desenvolvimento de redes neurais focadas em inspeção visual, por conta de suas múltiplas classes, permitindo a modelagem e o reconhecimento de padrões mais complexos.

Adicionalmente, selecionamos essa base de dados por possuir um caráter desafiador, uma vez que as imagens contidas em suas diferentes classes apresentam diferenças significativas em termos de posição do dano, tamanho e cor, tornando mais difícil a elaboração de um modelo eficiente.

Porém, ao lidarmos com um conjunto de dados que apresenta uma ampla variedade em suas imagens, estamos simulando condições mais realistas e complexas. Isso permite que a nossa rede neural seja treinada para lidar de forma eficiente e precisa com situações reais, onde as características dos danos podem variar consideravelmente. Ao enfrentar desafios como esses durante o treinamento, nossa rede neural terá a capacidade de generalizar melhor e responder de maneira adequada a novas imagens de danos em pontes, mesmo que apresentem características diferentes das imagens de treinamento.

No repositório original, os dados já estavam divididos na proporção de 70% para treino e 30% para teste, todavia optamos por unir ambas as partes em um único conjunto, em seguida aplicamos a divisão desse conjunto com base na validação cruzada estratificada, pois acreditamos que esta é uma forma mais apropriada de realizar a validação de nosso projeto em comparação a simples divisão do conjunto em teste e treino a partir desta proporção.

Para aumentar ainda mais a diversidade dos dados, empregamos a técnica de data augmentation para realizar transformações geométricas e de cor em nosso conjunto de dados. Utilizamos funções provenientes da biblioteca Keras para realizar alterações como: espelhar horizontalmente a imagem, aplicar zoom, ajustar o contraste, rotacionar a imagem e variar a intensidade do brilho. Além disso, utilizamos uma camada GaussianNoise() para introduzir ruído nos dados, gerando transformações adicionais de forma aleatória.

B. *Protocolo de Validação*

O protocolo de validação utilizado neste trabalho será a Validação Cruzada Estratificada. Este método de validação é uma variação do K-Fold Cross-Validation baseado em amostragem estratificada, que consiste em coletar as amostras na mesma proporção em que elas se encontram na população original. Ao aplicar essa técnica na seleção dos conjuntos de treino e teste, auxiliamos na generalização do modelo. Optamos por utilizar esse tipo de protocolo de validação devido à homogeneidade da distribuição dos nossos dados entre as classes. Ao usar a Validação Cruzada Estratificada, preservamos a distribuição original das classes, garantindo que o modelo seja exposto a todos os tipos de classe durante cada iteração do protocolo.

VII. Resultados/Discussões

Durante a execução do protocolo de validação cruzada estratificada, foi observado que o modelo da rede ResNet apresentou uma precisão média de 95% e uma perda de 0,08 no conjunto de treinamento.

No entanto, ao avaliar o desempenho do modelo em um conjunto de teste após cada treinamento, verificou-se uma queda na acurácia para cerca de 35% e um aumento na perda para 1,2. Esses resultados indicam claramente a ocorrência de overfitting, onde o modelo se ajustou excessivamente aos dados de treinamento, comprometendo sua capacidade de generalização.

Diante disso, decidimos reservar 20% do conjunto de treinamento para fins de validação. Essa abordagem nos permite avaliar o desempenho do modelo durante o treinamento em um conjunto de dados separado, que difere do conjunto de treinamento original.

Após dividir parte do conjunto para validação, treinamos o modelo da rede ResNet por 150 épocas, divididas em 30 épocas com camadas congeladas e 120 com parte das camadas descongeladas, obtendo o seguinte resultado:

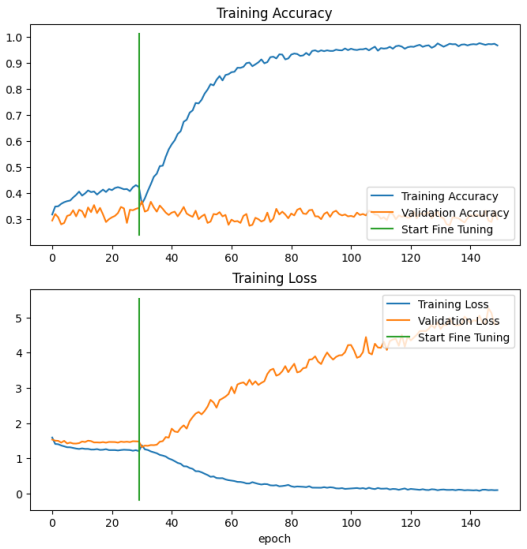

Fig 10. Gráfico de desempenho da rede ResNet101V2 com 4 classes.

Na figura 10, é perceptível que tanto a acurácia quanto a perda não apresentaram grandes variações durante o período em que as camadas estavam congeladas. No entanto, ao descongelar as camadas superiores, houve um notável avanço no conjunto de treinamento, com uma precisão de 98% e uma perda em torno de 0,08. Por outro lado, o conjunto de validação apresentou um aumento significativo na perda, enquanto a acurácia se manteve praticamente estável em cerca de 35%. É importante destacar que o treinamento da ResNet para 150 camadas, está levando em média 21 minutos.

Para evitar o overfitting, foram implementadas várias alterações, incluindo ajustes na taxa de aprendizado, aplicação de data augmentation, alteração do tamanho das imagens de entrada da rede, variação na quantidade de camadas descongeladas e a adição de camadas de dropout e regularização ativa. No entanto, apesar dessas modificações, não foi observada uma melhoria significativa nos resultados do conjunto de validação.

Analisando o conjunto de dados, observamos uma distribuição desigual das classes, em que a classe "Spalling" representa apenas 7% do conjunto de dados, enquanto a classe "Cracks" representa cerca de 37%. Percebendo que em alguns casos de teste o modelo prediz muito mais a classe "Cracks" do que as demais, temporariamente removemos a classe "Spalling" do conjunto de dados, a fim de verificar se esse desequilíbrio está ou não afetando negativamente o desempenho do modelo.

Após a remoção da classe "Spalling" do conjunto de dados, a acurácia do conjunto de validação apresentou crescimento, entretanto esse aumento provavelmente ocorreu devido ao fato de o modelo ter menos opções para escolher, o que aumenta a probabilidade de acertos aleatórios. Ao desenvolver a matriz de confusão para o conjunto com apenas 3 classes, obtemos a seguinte figura:

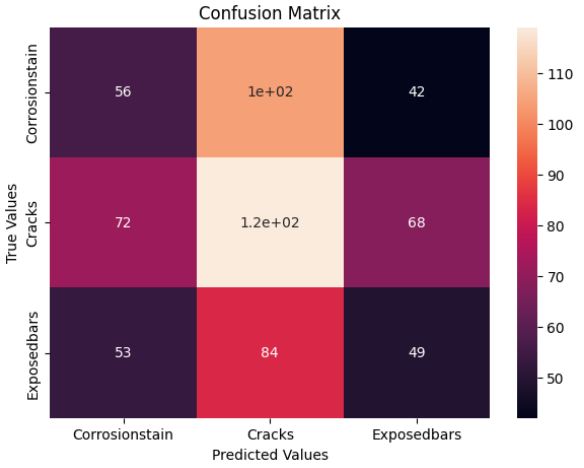

Fig 11. Matriz de confusão da rede ResNet com 3 classes.

Com base na figura 11, é perceptível que o modelo tende a prever com mais frequência a classe "Cracks" em comparação com as outras classes, principalmente ao analisar a classe "Corrosionstain", que foi frequentemente classificada como "Cracks", indicando uma confusão do modelo entre ambas as classes.

Além da rede ResNet, realizamos testes com as redes MobileNetV2, VGG16 e EfficientNetV2B1, utilizando os mesmos parâmetros.

A rede EfficientNetV2B1 alcançou uma média de 83% de precisão e 0,42 de perda para o conjunto de treinamento em 150 épocas. O tempo médio de treinamento para a EfficientNetV2B1 é de aproximadamente 11 minutos.

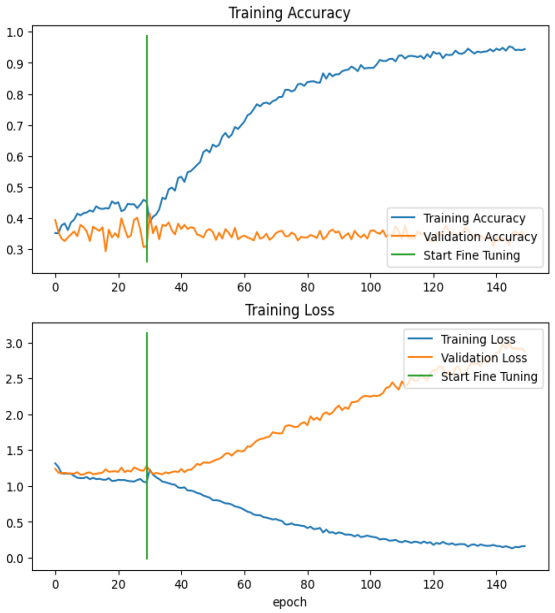

Fig 12. Gráfico de desempenho da rede EfficientNetV2B1 com 3 classes.

A rede MobileNetV2 mostrou-se uma das mais promissoras, alcançando uma média de 95% de precisão e 0,16 de perda para o conjunto de treinamento. Além disso, o

tempo médio de treinamento para a MobileNetV2 é de cerca de 7 minutos.

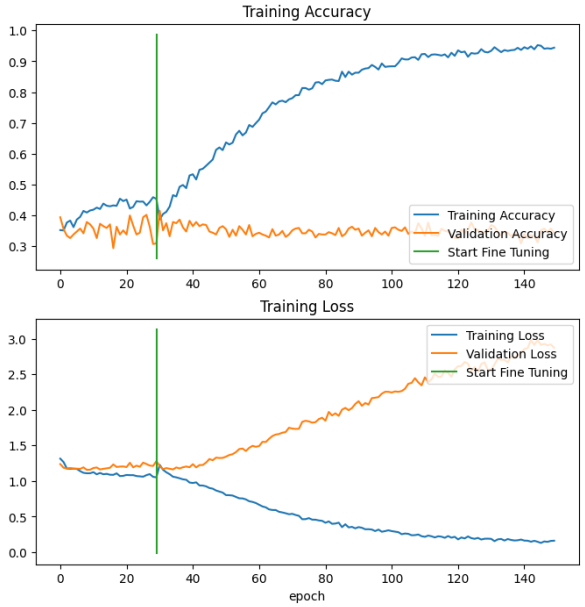

Fig 13. Gráfico de desempenho da rede MobileNetV2 com 3 classes.

A rede VGG16 demonstrou o pior desempenho entre todas as redes testadas. Ela obteve uma média de 56% de precisão e 0,92 de perda para o conjunto de treinamento. Além disso, o tempo médio de treinamento para a VGG16 foi de aproximadamente 19 minutos.

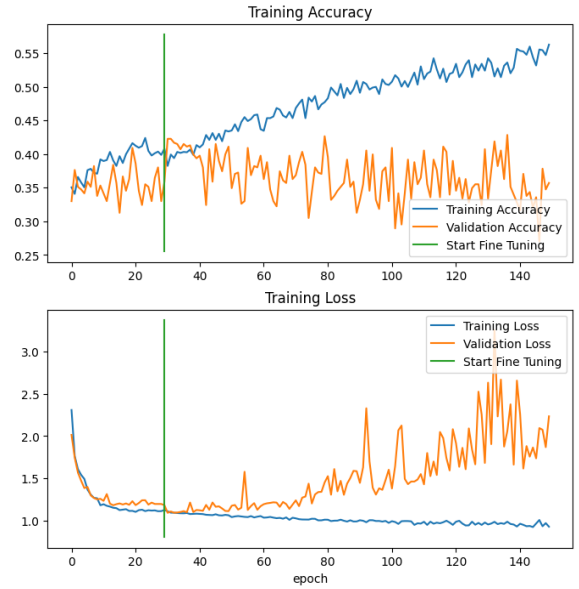

Figura 14. Gráfico de desempenho da rede VGG16 com 3 classes.

A tabela a seguir apresenta uma comparação de desempenho entre as redes:

	Acurácia Média Treino	Perda Média Treino	Acurácia Média Teste	Perda Média Teste	Tempo Médio em Minutos
ResNet	95%	0,08	35%	4,50	21
Efficient Net	83%	0,42	36%	1,80	11
Mobile Net V2	95%	0,16	33%	3,00	7
VGG 16	56%	0,92	34%	2,70	19
Inception V3	96%	0,09	35%	4,00	15
NASNet Mobile	97%	0,06	33%	4,50	23
XCeption	99%	0,02	31%	4,20	22

Fig 15. Tabela comparativa entre os resultados das redes.

Podemos perceber que as redes NASNet Mobile e XCeption apresentaram os melhores resultados para o conjunto de treino, entretanto, a rede MobileNet alcançou um ótimo resultado para o treino, com muito menos tempo que as outras redes, tornando-a umas das melhores redes com custo de tempo e benefício. Já para o conjunto de teste, a rede que apresentou os melhores resultados, apesar do overfitting, foi a rede EfficientNet, que assim como a MobileNet, não precisou de uma grande quantidade de tempo para ser treinada.

Assim, observamos que as duas redes que obtiveram os piores resultados no conjunto de treinamento, a VGG16 e a EfficientNet, apresentaram um desempenho superior no conjunto de teste. Esses resultados sugerem que, apesar de terem se saído pior durante o treinamento, essas redes foram mais eficientes em lidar com o overfitting, ainda que em uma escala pequena.

Como esses resultados não se apresentaram satisfatórios, continuamos a tentar resolver o problema de overfitting. Visto que as imagens de cada classe variam firmemente de tamanho, cor e posição espacial da anomalia, optamos por converter todas as imagens para escalas de cinza, visando reduzir a variação de imagem para imagem. Além disso, como a memória RAM do ambiente de execução não estava suportando corretamente nossa quantidade de imagens na rede, optamos por utilizar da técnica de undersampling e, reduzimos a quantidade de imagens das classes “Cracks” e “CorrosionStain” para 942 imagens, assim reduzimos a variação na quantidade de imagens entre as classes restantes e permitimos com que o ambiente de execução fosse capaz de analisar o modelo com imagens 224x224, haja vista que anteriormente, devido a limitação de RAM, era possível treinar o modelo apenas com imagens 160x160, essa quantidade maior de pixels permite com que o modelo possa extrair mais características durante o treinamento.

Essas alterações, somadas a mais algumas variações de hiperparâmetros como o batch size e a learning rate, nos permitiu alcançar valores de 41% de acurácia e 1,12 de perda em algumas instâncias, utilizando o modelo da MobileNet, que foi selecionado por ser o mais rápido dentre os modelos testados.

VIII. Conclusões e Trabalhos Futuros

Em suma, este trabalho cumpre o seu papel na elaboração de um algoritmo de aprendizado de máquina, que seja voltado para a inspeção visual automatizada. No entanto, o conjunto de dados abordado neste projeto é caracterizado por uma diversidade maior e menos padronização, englobando múltiplas classes, desigualdade na quantidade de imagens por classe e grandes variações em cores, tamanho e formato. Esses fatores exerceram influência na ocorrência de overfitting no modelo, uma vez que as imagens são tão distintas entre si que o modelo acaba aprendendo características específicas de cada imagem, em vez de padrões mais gerais. Isso resulta em um ajuste excessivo aos dados de treinamento.

Contudo, é evidente a robustez e o potencial do modelo desenvolvido neste trabalho, graças à sua base teórica sólida e aos resultados obtidos. Assim, o overfitting se torna um desafio que pode ser superado com êxito. Com ajustes apropriados, será possível melhorar a generalização do modelo e mitigar o problema de overfitting, aprimorando sua capacidade de lidar com conjuntos de dados desafiadores.

Como trabalho futuro, pretendemos aplicar e testar modelos de redes neurais mais recentes, uma vez que, para este estudo, nos concentramos nas redes clássicas. Ademais, com recursos computacionais adicionais, poderíamos explorar técnicas de oversampling para aumentar a amostra de dados e, consequentemente, mitigar o problema de overfitting identificado no conjunto de dados.

Além da utilização de redes neurais mais recentes, futuramente seria interessante explorar outras bases de dados. Como o BiNet era uma base de dados bastante complexa, é possível verificar a capacidade das redes em conjuntos mais simples, averiguando se ainda há overfitting. Por fim, com um modelo mais completo e sem muita influência do overfitting, podemos procurar realmente uma base de dados de anomalias em equipamentos de indústria, aplicando a IVA.

Referências Bibliográficas

[1] Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. Journal of industrial information integration, 6:1–10.

[2] Mitchell, T. (1997). Machine Learning. McGraw-Hill International Editions. McGrawHill.

[3] Gama, J., Faceli, K., Lorena, A., and De Carvalho, A. (2011). Inteligencia artificial: uma abordagem de aprendizado de máquina. Grupo Gen - LTC.

[4] de Padua Braga, A. (2007). Redes neurais artificiais: teoria e aplicações. LTC Editora.

[5] Hassaballah, M. and Awad, A. I. (2020). Deep learning in computer vision: principles and applications. CRC Press.

[6] Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press, Cambridge, MA, USA.

[7] Vargas, A. C. G., Paes, A., and Vasconcelos, C. N. (2016). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In Proceedings of the xxix conference on graphics, patterns and images, volume 1. Sn

[8] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

[9] Huang, S.-H. and Pan, Y.-C. (2015). Automated visual inspection in the semiconductor industry: A survey. Computers in industry, 66:1–10.

[10] Silva, H. E. S. et al. (2021). Classificação de lesões em imagens histológicas de mama usando wavelet e resnet-50.

[11] Silva, R. E. V. d. (2018). Um estudo comparativo entre redes neurais convolucionais para a classificação de imagens.

[12] He, T. Zhang, Z. Zhang H, Zhang, Z. X, Junyuan. Li, M. Bag of Tricks for Image Classification with Convolutional Neural Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 558-567

[13] Sandler, Mark et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv, 2019. Disponível em: https://arxiv.org/abs/1801.04381.

[14] Tsang, S.-H. (2019). Review: Nasnet — neural architecture search network (image classification). Disponível em: https://sh-tsang.medium.com/review-nasnet-neural-architecture-searchnetwork-image-classification-23139ea0425d.

[15] Liu, S., Wang, Q., & Luo, Y. (2019). A review of applications of visual inspection technology based on image processing in the railway industry. Transportation Safety and Environment, 1(3), 185-204. Disponível em: https://academic.oup.com/tse/article/1/3/185/5714252

[16] Seo, J., Duque, L., & Wacker, J. (2018). Drone-enabled bridge inspection methodology and application. Automation in construction, 94, 112-126. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0926580517309755.

[17] Ali, M. A., & Lun, A. K. (2019). A cascading fuzzy logic with image processing algorithm–based defect detection for automatic visual inspection of industrial cylindrical object's surface. *The International Journal of Advanced Manufacturing Technology*, *102*, 81-94. Disponível em: https://link.springer.com/article/10.1007/s00170-018-3171-7.

[18] Wang, Z., Li, T., Zheng, J. Q., & Huang, B. (2022). When CNN Meet with ViT: Towards Semi-Supervised Learning for Multi-Class Medical Image Semantic Segmentation. Disponível em: https://arxiv.org/abs/2208.06449.

[19] DataScienceTeam (2020). Uma visao geral da resnet e suas variantes. Disponível em: https://datascience.eu/pt/aprendizado-de-máquina/uma-visão-geral-da-resnet-e suas-variantes/.

[20] Martínez, F., and Jacinto, E. (2020). Performance evaluation of the nasnet convolutional network in the automatic identification of covid-19. International Journal on Advanced Science, Engineering and Information Technology, 10(2):662.

[21] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. Disponível em: https://arxiv.org/abs/2010.11929.

[22] Bukhsh, Z.A, Bridge visual inspection dataset and approach for damage detection. (2022). GitHub repository. Disponível em: https://github.com/Zaharah/BiNet-bridge-visual-inspection-dataset

[23] ul Hassan, M. (2018). Vgg16-convolutional network for classification and detection. Neurohive.

[24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[25] Nogra, J. A. (2021). Detecting face mask usage using inception network.

[26] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.