Redes Neurais Profundas com Saídas Antecipadas para Imagens com Distorção em Ambientes de Nuvem

Roberto G. Pacheco, Fernanda D.V.R. Oliveira, Rodrigo S. Couto *

[1]Universidade Federal do Rio de Janeiro - GTA/PADS/PEE-COPPE/DEL-Poli

{pacheco, rodrigo}@gta.ufrj.br, fernanda.dvro@poli.ufrj.br

Abstract. *Deep neural networks (DNNs) are sensitive to distorted images, reducing their accuracy. This work analyzes how solutions of early-exit DNNs (EE-DNNs) can solve this problem. EE-DNNs have side branches inserted into their middle layers to classify images earlier, at the edge, and thus avoid sending them to the cloud. Besides, multiple side branches can compose an ensemble that collectively produces a more accurate inference. The results, in terms of accuracy, show that EE-DNNs and the ensemble are as sensitive as conventional DNNs. However, given the edge usage, these approaches can reduce the inference time and the number of operations to classify images.*

Resumo. *As redes neurais profundas (DNNs) são sensíveis a imagens com distorção, tendo sua acurácia reduzida. Este trabalho analisa como soluções de DNNs com saídas antecipadas (EE-DNNs) podem resolver esse problema. As EE-DNNs possuem ramos laterais inseridos em suas camadas intermediárias para classificar antecipadamente amostras na borda e evitar envios para a nuvem. Além disso, múltiplos ramos laterais podem compor um comitê e produzir coletivamente uma inferência mais precisa. Os resultados, em termos de acurácia, mostram que EE-DNNs e o comitê são tão sensíveis quanto DNNs convencionais. Entretanto, dado o uso da borda, essas abordagens conseguem reduzir o tempo de inferência e o número de operações para classificar imagens.*

1. Introdução

As Redes Neurais Profundas (*Deep Neural Networks* – DNNs) têm obtido progressos em aplicações de visão computacional, como classificação de imagens e detecção de objetos. Contudo, diversos trabalhos demonstraram experimentalmente que o desempenho das DNNs é altamente sensível à presença de distorção na imagem [Dodge e Karam, 2016, Pacheco et al., 2021c, Dodge e Karam, 2018, Liu et al., 2020]. Essas distorções podem ser consequência de condições externas, como movimentação no momento da captura ou fatores ambientais [Liu et al., 2020]. Dessa forma, as DNNs sofrem um impacto significativo em sua acurácia ao classificar imagens com diferentes tipos de distorções. Portanto, é necessário desenvolver técnicas e estratégias para tornar as DNNs mais robustas a distorções nas imagens.

Este trabalho visa analisar se soluções baseadas em DNNs com saídas antecipadas (*Early-Exit* DNNs – EE-DNNs) são capazes de tornar a inferência mais robusta

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. O trabalho também foi financiado pelo CNPq, FAPERJ (E-26/010.002174/2019 e E-26/201.300/2021), PR2/UFRJ, e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), auxílio no. 2015/24494-8.

contra distorções nas imagens em comparação a DNNs convencionais. As EE-DNNs consistem em DNNs na quais são inseridos ramos laterais ao longo de sua arquitetura [Teerapittayanon et al., 2016, Laskaridis et al., 2020]. Dada uma imagem como entrada, os ramos laterais são capazes de estimar a confiança da classificação dessa imagem considerando somente as camadas da DNN anteriores ao ramo. Caso o valor da confiança seja maior do que um dado limiar, a imagem é considerada suficiente confiável e, consequentemente, pode ser classificada antecipadamente no ramo. As EE-DNNs baseiam-se no fato de que as imagens capturadas não possuem o mesmo nível de dificuldade de classificação e que uma parcela significativa de imagens pode ser classificada antecipadamente nos ramos laterais. Dessa forma, reduz-se o número de camadas neurais a serem processadas e, portanto, o atraso de processamento. Uma outra abordagem é empregar múltiplos ramos laterais para compor um comitê (*ensemble*) [Qendro et al., 2021]. Nesse caso, usam-se as inferências fornecidas individualmente por cada ramo lateral com objetivo de produzir coletivamente uma única inferência. A partir dessa inferência colaborativa, classifica-se a imagem.

A análise deste trabalho visa verificar se as EE-DNNs e a estratégia de comitê de ramos laterais tornam a inferência mais robusta contra imagens com *blur* gaussiano em comparação a DNNs convencionais. Para isso, este trabalho implementou as EE-DNNs e o comitê de ramos laterais em um cenário *offloading* adaptativo, no qual parte da EE-DNN é processada no dispositivo em borda. Caso a inferência provida pelos ramos laterais não atinjam um dado limiar de confiança, o dispositivo em borda transmite a inferência a nuvem que executa a outra parte da EE-DNN. Este trabalho demonstra experimentalmente que a utilização de EE-DNNs e comitê de ramos laterais conseguem atingir um nível de acurácia equivalente a DNNs convencionais para imagens com diversos níveis de *blur*. Além disso, dado um cenário de *offloading* adaptativo auxiliado por EE-DNNs, este trabalho mostrou que EE-DNNs e comitê de ramos laterais consegue reduzir significativamente o tempo de inferência e o número de operações com ponto flutuante (*FLoating-point Operations Per Second* – FLOPS), em razão de classificar uma quantidade expressiva de imagens antecipadamente na borda.

Este trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados existentes na literatura. A Seção 3 introduz o conceito de EE-DNNs. A Seção 4.1 apresenta o conjunto de dados empregado e o tipo de distorção aplicada nas imagens. A Seção 4 analisa como soluções de DNN de saídas antecipadas se comportam ao receberem imagens distorcidas. Por fim, a Seção 5 conclui este trabalho e propõe os próximos passos.

2. Trabalhos relacionados

Dodge e Karam demonstram experimentalmente que as DNNs são sensíveis à distorção da imagem [Dodge e Karam, 2016]. Logo, imagens com distorção degradam significativamente o desempenho de DNNs. Nesse sentido, diferentes artigos propõem estratégias para tornar as DNNs convencionais mais robustas às imagens com distorção.

Rozsa *et al.* propõem um método de treinamento de DNN que provê robustez especificamente contra imagens com ruído [Rozsa et al., 2016]. Contudo, essa proposta atende a um único tipo de distorção. Para resolver tal limitação, Dodge e Karam propõem uma combinação de DNNs especialistas, de modo que cada DNN é treinada somente para um

tipo específico de distorção [Dodge e Karam, 2018]. A saída fornecida por cada modelo especialista é combinada por meio de uma média ponderada, atuando como um comitê de modelos. Similarmente, Collabar [Liu et al., 2020] também treina um modelo de DNN especialista para cada tipo de distorção. Porém, Collabar implementa um classificador de distorção para identificar o tipo de distorção presente na imagem e selecionar a DNN especialista apropriada. Contudo, a desvantagem dessas propostas é a necessidade de armazenar e executar um modelo especialista para cada possível tipo de distorção na imagem. Consequentemente, essa proposta pode ser inviável de ser implementada principalmente em dispositivos de borda limitados em termos de memória e processamento. Para resolver esse problema, Pacheco *et al.* propõem uma arquitetura de EE-DNNs com ramos especialistas, na qual apenas os ramos laterais são treinados especificadamente para cada tipo de distorção [Pacheco et al., 2021c]. Portanto, a proposta de EE-DNNs com ramos especialistas pode escalar o número de tipos de distorção a serem consideradas.

Diferentemente das propostas anteriores, este trabalho não foca desenvolver novas estratégias que torne a DNN ou a EE-DNN especialistas em tipos específicos de distorção. Assim, este trabalho não se propõe em treinar modelos específicos para cada tipo de distorção, como em [Liu et al., 2020] e [Dodge e Karam, 2018], nem treinar ramos laterais para que se tornem especialistas em uma distorção específica, como em [Pacheco et al., 2021c]. Pelo contrário, este trabalho visa comparar o emprego de soluções baseada em EE-DNNs a DNNs convencionais para melhorar o desempenho em classificar imagens com distorção. Dessa forma, este trabalho não treina os ramos laterais em um tipo de distorção específica. Este artigo propõe então uma análise com o objetivo de recomendar que aplicações as quais considerem receber imagens com distorções empreguem minimamente soluções de EE-DNNs em detrimento a DNNs convencionais. Além disso, diferente das propostas anteriores, à exceção de[Pacheco et al., 2021c], este trabalho implementa um cenário de *offloading* adaptativo para reduzir o tempo de inferência em comparação com uma DNN convencional processada apenas na nuvem. Em [Pacheco e Couto, 2020] também se considera um cenário de *offloading* adaptativo com imagens distorcidas. Entretanto, o seu foco é escolher em qual camada dividir o modelo de DNN entre a borda e a nuvem. Assim, diferentemente deste trabalho, não analisa com detalhes o desempenho das soluções e não considera uma estratégia de comitê. Além disso, em [Pacheco e Couto, 2020] o cenário de *offloading* adaptativo é apenas emulado. Por outro lado, este trabalho utiliza uma solução completa de *offloading* adaptativo, já utilizada em outros trabalhos [Pacheco et al., 2021a, Pacheco et al., 2021c], e hardware realista, como um Nvidia Jetson Nano na borda e uma máquina virtual na Amazon AWS EC2 na nuvem.

Na literatura, há diversos artigos sobre EE-DNNs. A BranchyNet [Teerapittayanon et al., 2016] é uma proposta de EE-DNN que utiliza a entropia para decidir se uma determinada amostra pode ser classificada antecipadamente. Diferentemente, o SPINN [Laskaridis et al., 2020] toma tal decisão baseado em uma estimativa de confiança da classificação. Os trabalhos da BranchyNet e do SPINN demonstram experimentalmente que, em diferentes *datasets* de classificação de imagens, uma parcela significativa das imagens pode ser classificada antecipadamente. Dessa forma, o principal objetivo desses trabalhos é reduzir o tempo de inferência, ou seja, o tempo necessário para classificar uma dada imagem. Outras propostas, como o FlexDNN [Fang et al., 2020] e o Edgent [Li et al., 2019], empregam EE-DNNs para

selecionar a profundidade (isto é, a quantidade de camadas neurais a serem processadas) mais adequada para alcançar um requisito de acurácia e de tempo de inferência. Por outro lado, há outra direção de trabalhos que objetivam implementar EE-DNNs em diferentes *hardwares*. Dynexit [Wang et al., 2019] treina e implementa EE-DNN em placas FPGA (*Field-Programmable Gate Array*) para reduzir o tempo de inferência.

Este artigo investiga o impacto de imagem com distorções na inferência de EE-DNNs, considerando um cenário de *offloading* adaptativo. Todos os trabalhos anteriores de EE-DNNs, exceto [Pacheco et al., 2021c] e [Pacheco e Couto, 2020], assumem implicitamente que as amostras recebidas pelo modelo são imagens sem qualquer distorção, o que não condiz com aplicações reais. Por exemplo, um dispositivo móvel, como *smartphone*, pode capturar imagens com diferentes tipos de distorção, como o *blur*, e com diferentes níveis de distorção [Secci e Ceccarelli, 2020].

3. Redes neurais profundas com saídas antecipadas

As redes neurais profundas (DNNs) podem ser descritas como uma sequência de camadas neurais capazes de extrair os atributos e características das imagens de entrada e aprender por meio desses atributos. Em geral, as primeiras camadas neurais das DNNs extraem atributos mais simples, enquanto as camadas neurais mais profundas são responsáveis por extrair atributos mais complexos. As redes neurais profundas com saídas antecipadas (EE-DNNs) partem do princípio de que diferentes amostras podem necessitar de atributos com diferentes níveis de complexidade para serem classificadas de forma adequada [Teerapittayanon et al., 2016]. Em outras palavras, há amostras do conjunto de dados que podem ser facilmente classificadas usando atributos extraídos pelas primeiras camadas neurais. Portanto, tais amostras podem ser classificadas logo após o processamento dessas primeiras camadas neurais. Para isso, as EE-DNNs possuem ramos laterais inseridos em suas camadas intermediárias. Por outro lado, há outras amostras que necessitam de atributos mais complexos obtidos pelas camadas neurais mais profundas. Nesse caso, essas amostras precisam ser processadas pelas camadas profundas para poderem obter uma classificação acurada.

A Figura 1 ilustra uma EE-DNN com múltiplos ramos laterais inseridos ao longo de sua arquitetura. Nessa figura, os vértices $L_1, \cdots, L_N$ são as camadas neurais (*layers*) do ramo principal (*backbone*) da DNN. Os ramos laterais (*branches*) são representados pelos vértices $b_1, \cdots, b_k$. Esses ramos laterais b_i são capazes de classificar determinadas amostras a partir dos atributos extraídos pelas camadas neurais do ramo principal anteriores ao ramo lateral em questão $L_1, \cdots, L_i$. Este trabalho insere três ramos laterais de forma equidistante em termos de FLOPS ao longo da arquitetura do ramo principal da DNN, seguindo a metodologia desenvolvida por SPINN [Laskaridis et al., 2020]. Dessa forma, a EE-DNN utilizada possui quatro saídas, incluindo a camada de saída presente no ramo principal. Em relação ao treinamento, este trabalho emprega o procedimento tradicional de treinamento apresentado por [Teerapittayanon et al., 2016] e [Laskaridis et al., 2020].

3.1. *Offloading* adaptativo

Este trabalho implementa a EE-DNN em um cenário de *offloading* adaptativo [Pacheco et al., 2021a]. A Figura 2 ilustra esse cenário, no qual uma parte da EE-DNNs é implementada no dispositivo de borda, enquanto as camadas restantes são processadas no servidor em nuvem. Primeiramente, define-se o número k de ramos laterais,

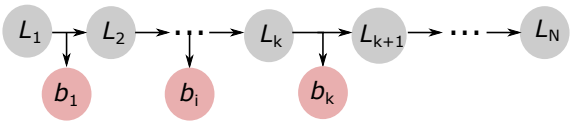


Figura 1. Arquitetura genérica de uma EE-DNN.

que executam no dispositivo em borda. O resultado da camada L_k é enviado para a nuvem caso os ramos $b_1, \cdots, b_k$ não realizem a classificação.

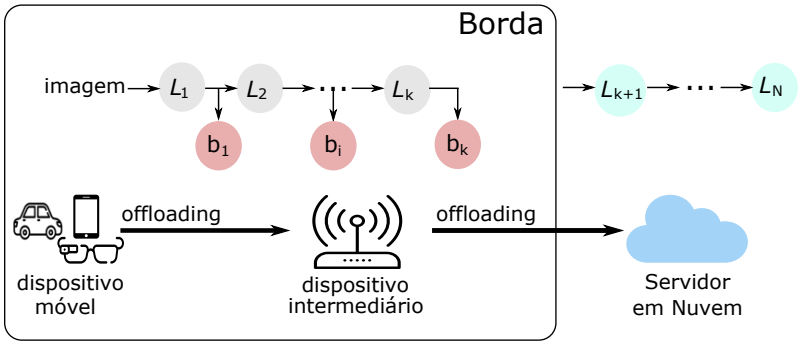


Figura 2. Ilustração do cenário de *offloading* adaptativo auxiliado por EE-DNNs. Adaptada de [Pacheco et al., 2021c]

Uma vez que a EE-DNN foi treinada, esse modelo recebe uma imagem de entrada e a processa para realizar a inferência e classificar o objeto presente na imagem. O dispositivo em borda processa a imagem por cada camada neural até atingir o ramo lateral b_i, onde $i = 1, \cdots, k$, sendo k o número de ramos laterais. Nesse momento, o ramo lateral b_i gera o vetor de saída $\mathbf{z}_i$. A partir desse vetor $\mathbf{z}_i$, é possível obter o vetor de probabilidade $\boldsymbol{p}_i = \text{softmax}(\boldsymbol{z}_i) \propto \exp(\boldsymbol{z}_i)$, cujo valor de cada elemento corresponde à probabilidade de a imagem pertencer a uma determinada classe. Em seguida, calcula-se a confiança de classificação f_i como sendo $f_i = \max_{c \in \mathcal{C}} p_{i,c}$, no qual $p_{i,c}$ é o c-ésimo elemento do vetor $\boldsymbol{p}_i$ e $\mathcal{C}$ é o conjunto das possíveis classes. Se o valor da confiança f_i for maior ou igual a um determinado limiar α, isto é, $f_i \geq \alpha$, o dispositivo em borda conclui a inferência e o ramo lateral b_i pode classificar a imagem como $\hat{y} = \arg\max_{c \in \mathcal{C}}(p_{i,c})$, no qual $\hat{y}$ denota o rótulo classificado pelo ramo lateral. Caso contrário, isto é, $f_i < \alpha$, a amostra segue sendo processada pelas próximas camadas do ramo principal até alcançar o próximo ramo lateral b_{i+1}, no qual o procedimento descrito anteriormente repete-se. Se nenhum dos ramos laterais alocados na borda for capaz de atingir o limiar α, então o dispositivo da borda envia a inferência à nuvem que processa o restante das camadas neurais do ramo principal da EE-DNN.

3.2. Comitê de ramos laterais com *offloading* adaptativo

A estratégia de comitê de ramos laterais visa utilizar as inferências fornecidas individualmente por cada ramo lateral para produzir coletivamente uma única inferência ainda mais acurada [Qendro et al., 2021]. Ou seja, a inferência nunca termina nos ramos laterais, mas seus resultados são acumulados para formar um comitê. Dessa forma, após processar os k ramos laterais, obtém-se o conjunto de vetores de probabilidades dado por $\mathcal{K} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k\}$.

Após processar k ramos laterais, Qendro *et al.* obtém a inferência total $\boldsymbol{p}_{\mathcal{E}}(\mathcal{K})$ produzida pelo comitê de ramos laterais, a partir da média aritmética do vetor de probabilidades p_i dos ramos laterais [Qendro et al., 2021]. Contudo, este trabalho aplica uma média ponderada, na qual a ponderação é feita pela acurácia de cada ramo b_i denotada por Acc_i. O objetivo de empregar a média ponderada é que a influência de um ramo na inferência total, produzida pelo comitê de ramos laterais, seja proporcional à sua acurácia obtida no conjunto de validação. É válido notar que, neste artigo, a inferência total dada pelo comitê não considera a inferência provida pela camada de saída do ramo principal, em razão de estar implementado na nuvem. Durante a inferência, a imagem de entrada é processada por cada camada neural, incluindo os k ramos laterais implementados na borda, resultando no conjunto de inferências $\mathcal{K}$. A inferência total produzida pelo comitê de ramos lateral $\boldsymbol{p}_{\mathcal{E}}(\mathcal{K})$ é calculada como uma média ponderada dada por

$$\boldsymbol{p}_{\mathcal{E}}(\mathcal{K}) = \sum_{i=1}^{k} \text{Acc}_i \cdot \boldsymbol{p}_i. \tag{1}$$

Dessa forma, a Equação 1 utiliza a acurácia de cada ramo lateral Acc_i como peso para calcular o vetor de probabilidade $\boldsymbol{p}_{\mathcal{E}}(\mathcal{K})$ como uma média ponderada. Como na Seção 3.1, utiliza-se esse vetor para obter a confiança da classificação $f_{\mathcal{E}}(\mathcal{K}) = \max_{c \in \mathcal{C}} p_{\mathcal{E},c}(\mathcal{K})$, no qual $p_{\mathcal{E},c}$ é c-ésimo elemento do vetor $\boldsymbol{p}_{\mathcal{E}}(\mathcal{K})$. Em seguida, verifica-se se $f_{\mathcal{E}}(\mathcal{K}) \geq \alpha$. Nesse caso, o dispositivo em borda classifica a imagem como $\hat{y} = \arg\max_{c \in \mathcal{C}}(p_{\mathcal{E},c}(\mathcal{K}))$. Caso contrário, o dispositivo em borda envia a inferência à nuvem para processar o ramo principal, como explicado na Seção 3.1.

4. Experimentos

Os experimentos deste trabalho avaliam o impacto da adoção de soluções baseadas em DNNs com saídas antecipadas para classificar imagens com diferentes níveis de *blur* em comparação a DNN convencional. Para isso, compara-se o desempenho de classificação usando o comitê de saídas antecipadas com o desempenho obtido por uma DNN convencional implementada apenas no servidor em nuvem. O uso da DNN convencional corresponde a uma estratégia tradicional, no qual o dispositivo móvel coleta a imagem e a transmite à nuvem. Esta seção descreve o conjunto de dados e o cenário experimental utilizado e, em seguida, apresenta os experimentos. Todo código desenvolvido está disponível em um repositório aberto[1].

4.1. Conjunto de dados utilizado

Este trabalho utiliza o conjunto de dados Caltech-256 [Griffin et al., 2007], que contém 30.607 imagens coloridas, sem distorção, de objetos cotidianos, como motocicletas, bandeiras e tênis. Esse conjunto de dados possui 257 categorias e foi dividido em 80% das imagens para o conjunto de treinamento, 10% para o conjunto de validação e 10% para o conjunto de teste.

Para este trabalho, é necessário aplicar um procedimento para inserir diferentes níveis de distorção nas imagens. A distorção avaliada neste trabalho é o *blur* gaussiano. Uma imagem com *blur* gaussiano pode ser modelada da seguinte forma: $g(\boldsymbol{x}|\sigma_{GB}) =$

[1] `https://github.com/pachecobeto95/DR_EENN`

$f(\boldsymbol{x}) * h(\boldsymbol{x}|\sigma_{GB})$, sendo $\boldsymbol{x} = (x_1, x_2)$ as coordenadas do pixel. O termo $g(\boldsymbol{x}|\sigma_{GB})$ representa a imagem distorcida com *blur* gaussiano, enquanto $f(\boldsymbol{x})$ denota a imagem sem distorção e $h(\boldsymbol{x}|\sigma_{GB})$ trata-se do filtro gaussiano. O desvio padrão desse filtro é dado por σ_{GB}. O nível nível de *blur* presente na imagem depende do desvio padrão do filtro gaussiano σ_{GB}. Assim, quanto maior for σ_{GB}, mais a imagem é distorcida com *blur*. Este trabalho emprega $\sigma_{GB} \in \{0, 0.1, 0.2, 0.5, 0.8, 0.9, 1, 1.2, 1.5, 1.8\}$. As imagens sem distorção correspondem a $\sigma_{GB} = 0$.

4.2. Cenário experimental

Este trabalho reproduz um cenário realista de *offloading* adaptativo, apresentado na Figura 2. Para tal, emprega-se uma placa Nvidia Jetson Nano[2] como dispositivo em borda que executa o sistema operacional (SO) Ubuntu 18.04 e está localizado no Rio de Janeiro, Brasil. Esse dispositivo em borda está equipado com uma GPU Nvidia Maxwell com 128 núcleos e uma CPU ARM A57 com quatro núcleos a 1.43 GHz. Além disso, o experimento utiliza uma máquina virtual da Amazon AWS EC2[3] (*Elastic Cloud Computing*) como o servidor em nuvem, instanciado em São Paulo, Brasil. Essa cidade é escolhida por seu o sítio da Amazon AWS mais próximo ao Rio de Janeiro, analisando o melhor caso de uso da nuvem. A máquina da nuvem é uma instância `g4dn.xlarge` com quatro vCPUs da Intel Cascade Lake e uma GPU NVIDIA Tesla T4 executando um SO Ubuntu 20.04 LTS. Para implementar o cenário, incluindo a aplicação no servidor em nuvem, utiliza-se Python Flask[4]. O dispositivo de borda está conectado à Internet por meio de uma interface de rede Ethernet Gigabit e se comunica com o servidor em nuvem por HTTP.

Antes de executar os experimentos, verificam-se as condições de rede entre o dispositivo de borda, Nvidia Jetson Nano, e o servidor em nuvem, a instância `g4dn.xlarge` da Amazon EC2. Para isso, mede-se o RTT (*Round Trip Time*) e a vazão, empregando as ferramentas `ping` e `iPerf3` respectivamente. O valor de médio de vazão obtido é de 93,9 Mbps, com desvio padrão de 3,8 Mbps. O RTT médio é de 11.91 ms, com desvio padrão de 0.66 ms. É importante notar que esses valores de vazão e RTT são usados apenas ilustrar as condições da rede, já que tais condições podem variar ao longo do experimento.

Este trabalho utiliza a MobileNetV2 [Howard et al., 2017] como a arquitetura da DNN convencional, em razão de ser um modelo leve e desenvolvido para ser executado em em dispositivo com limitação de recursos computacionais. Em relação à EE-DNN, insere-se três ramos laterais em camadas intermediárias ao longo da arquitetura da MobileNetV2, adotando a estratégia proposta por SPINN [Laskaridis et al., 2020]. Conforme visto na Seção 3, as EE-DNNs possuem um limiar de confiança α para decidir se uma imagem deve ser classificada antecipadamente. Adota-se $\alpha \in \{0.8, 0.9\}$ nos experimentos. As DNN e EE-DNNs utilizadas são calibradas usando a metodologia descrita em [Pacheco et al., 2021a].

Para cada rodada experimental, adiciona-se *blur* gaussiano com um dado nível, que varia de acordo com σ_{GB}, em cada uma das imagens do conjunto de dados. Na execução do experimento com as EE-DNNs e o comitê, primeiramente define-se o número de ramos laterais ativos (isto é, ramos que podem terminar a inferência) na borda e o limiar de

[2]https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

[3]https://aws.amazon.com/ec2/

[4]https://flask.palletsprojects.com/en/2.2.x/

confiança α. Em seguida, o dispositivo de borda inicia um contador para medir o tempo de inferência e executa o processamento das camadas neurais da borda. Se a classificação atingir os critérios para terminar a classificação na borda, como visto na Seção 3, o contador é finalizado. Caso contrário, o dispositivo em borda envia os dados para o servidor da nuvem, que, por sua vez, processa o restante das camadas neurais da DNN. Nesse caso, o dispositivo de borda aguarda por uma resposta HTTP vinda do servidor em nuvem para finalizar o contador.

Na execução do experimento com a DNN convencional, primeiramente inicia-se um contador e imediatamente envia-se a imagem ao servidor em nuvem, sem processar nenhuma das camadas na borda. Ao receber a imagem, a nuvem processa todas as camadas neurais da DNN convencional. Como descrito anteriormente, o dispositivo de borda aguarda por uma resposta HTTP vinda da nuvem para finalizar o contador.

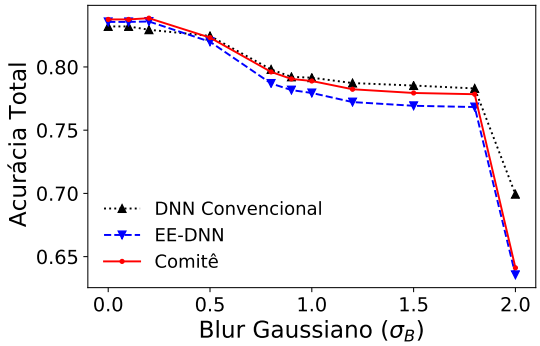
Após concluir uma rodada experimental, é possível obter o tempo de inferência necessário para classificar uma amostra e avaliar se a classificação realizada foi correta. O procedimento descrito é realizado para cada uma das imagens do conjunto de teste. Dessa forma, obtém-se acurácia total medida pelo modelo, além do tempo de inferência médio. Além disso, no caso de EE-DNN e comitê de ramos laterais, avalia-se também a probabilidade de classificação antecipada na borda. As próximas seções apresentam tais resultados comparando a EE-DNN sem o comitê, que serão tratadas simplesmente por "EE-DNN", com o comitê de saídas antecipadas, tratadas por "comitê", e a DNN convencional.

4.3. Acurácia

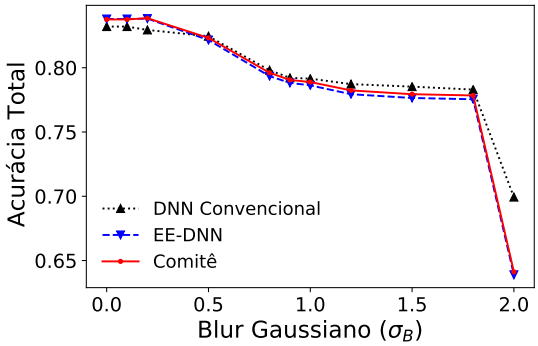
Nesta seção avalia-se a acurácia total obtida pela EE-DNN, pelo comitê de ramos laterais e pela DNN convencional para diferentes níveis de *blur* gaussiano. Dado um nível de distorção, a acurácia total é calculada como o número de imagens classificadas corretamente dividido pelo número total de imagens, considerando tanto imagens classificadas no dispositivo de borda, quanto no servidor em nuvem.

A Figura 3 mostra os resultados de acurácia total em função do nível de *blur*, considerando diferentes limiares de confiança α e número de ramos ativos na borda. Note que os resultados para a DNN convencional são iguais em todas as figuras, pois não dependem de α e do número de ramos. À primeira vista, a figura mostra que a acurácia dos modelos avaliados, EE-DNN, comitê e DNN convencional, reduzem à medida que aumenta o nível de *blur* presente nas imagens. Esse resultado corrobora a conclusão de [Dodge e Karam, 2016], de que a distorção nas imagens degrada a acurácia das DNNs. A figura também mostra que, para imagens sem distorção ou com baixo nível de distorção (isto é, $\sigma_{GB} \in \{0, 0.1, 0.2\}$), a acurácia da EE-DNN e do comitê de ramos laterais consegue até mesmo superar o desempenho da DNN convencional. Esse resultado já é esperado, em razão da EE-DNN se adaptar a imagens com diferentes complexidades de classificação dentro de um mesmo conjunto de dados [Teerapittayanon et al., 2016, Pacheco et al., 2021b]. O interessante desses resultados é observar que o comitê de saídas antecipadas obteve desempenho similar à EE-DNN e permaneceu com desempenho maior que a DNN convencional após inserir baixo nível de *blur*. Portanto, esses resultados recomendam, mesmo para imagens sem distorção ou com baixo nível de *blur*, a utilização de soluções baseadas em EE-DNNs em ao invés de uma DNN convencional.

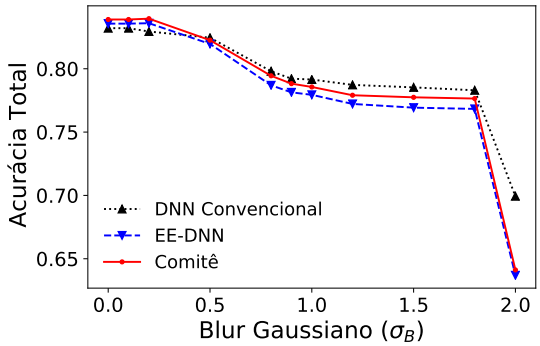
Ao comparar o desempenho de EE-DNNs e do comitê de ramos laterais na Figura 3, nota-se que o comitê obteve um ligeiro ganho de acurácia em relação à EE-DNN. Esse comportamento ocorre pois o comitê de ramos laterais utiliza múltiplos ramos para obter uma estimativa de confiança de classificação $f_{\mathcal{E}}(\mathcal{K})$ mais precisa do que as estimativas de confiança fornecida por cada ramo individualmente. Assim, o comitê aprimora as decisões em relação às amostras que devem ser classificadas na borda. Consequentemente, o comitê classifica menos imagens antecipadamente na borda, como será apresentado na Seção 4.4, recorrendo mais vezes à DNN mais robusta na nuvem. Isso torna o comitê uma solução intermediária entre a EE-DNN e a DNN convencional. Apesar do ligeiro aumento de acurácia em relação à EE-DNN, o comitê apresenta desvantagens em comparação à EE-DNN, como serão detalhadas nos próximos experimentos. Como o comportamento da acurácia para três ramos é similar ao de dois, os demais experimentos deste trabalho consideram que todos os três ramos estão ativos.



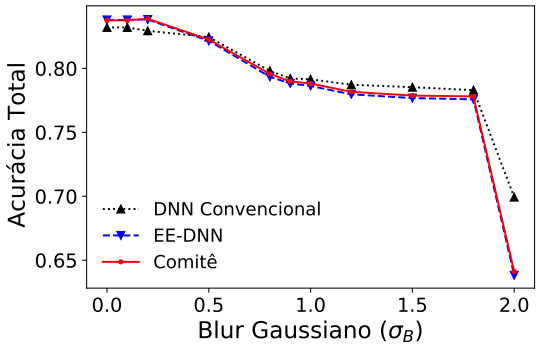
(a) Dois ramos laterais e $\alpha = 0.8$.



(b) Dois ramos laterais e $\alpha = 0.9$.



(c) Três ramos laterais e $\alpha = 0.8$.



(d) Três ramos laterais e $\alpha = 0.9$.

Figura 3. Acurácia total da EE-DNN, de ramos laterais e da DNN convencional, considerando diferentes limiares de confiança e diferentes números de ramos laterais implementados na borda.

4.4. Probabilidade de classificação antecipada

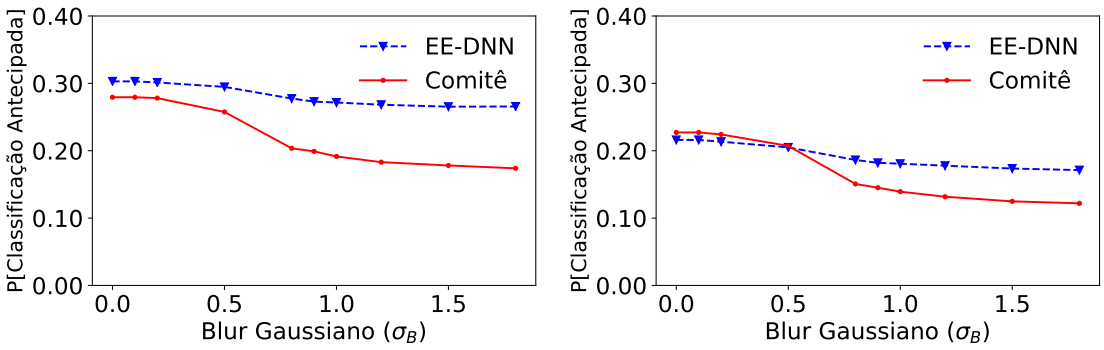
Esta seção avalia o impacto das EE-DNNs e a adoção da estratégia de comitê de ramos laterais em termos da probabilidade de classificar uma imagem no dispositivo de borda. Para obter essa probabilidade, primeiramente executa-se o procedimento experimental descrito na Seção 4.2. Em seguida, basta calcular essa probabilidade como sendo o número de amostras classificadas antecipadamente no dispositivo em borda dividido pelo

número total de amostras presentes no conjunto de teste. O cálculo dessa probabilidade é realizado para cada nível de *blur* gaussiano.

A Figura 4 mostra os resultados da probabilidade de classificação na borda em função de diferentes níveis de *blur* gaussiano usando três ramos laterais na borda. A curva azul corresponde aos resultados obtidos pela EE-DNN, enquanto a curva vermelha refere-se aos resultados obtidos pelo comitê de ramos laterais. Note que não há resultados da DNN convencional, visto que a probabilidade de classificação na borda é sempre zero.

Primeiramente, a Figura 4 mostra que uma parcela significativa de imagens do conjunto de teste pode ser classificada antecipadamente no dispositivo de borda. Por exemplo, a Figura 4(a) mostra que, pelo menos, 20% das imagens com *blur* gaussiano de $\sigma_{GB} = 1$ podem ser classificadas antecipadamente, independentemente do uso de comitê ou não. Esse resultado valida a implementação de soluções baseadas em EE-DNNs com imagens distorcidas, pois uma parcela significativa pode ser classificada na borda.

A Figura 4 mostra que o comitê reduz o número de imagens classificadas antecipadamente na borda em comparação à EE-DNN para diversos níveis de *blur*. Dessa forma, os resultados das Figuras 3 e 4 mostram um claro compromisso entre acurácia e a probabilidade de classificação antecipada, pois o comitê necessita recorrer mais à nuvem para obter um ligeiro ganho de acurácia em relação ao EE-DNN. Assim, espera-se que, em comparação com a EE-DNN, o comitê aumente o tempo de inferência dada a necessidade de enviar mais imagens à nuvem. Além disso, o comitê necessita de mais processamento pois todos os ramos são utilizados na inferência. Os próximos experimentos focam analisar essa penalidade na necessidade de processamento e tempo de inferência na estratégia do comitê, além de comparar os resultados com a DNN convencional.



(a) $\alpha = 0.8$. (b) $\alpha = 0.9$.

Figura 4. Probabilidade de classificação antecipada na borda de EE-DNNs e comitê de ramos laterais, considerando diferentes limiares de confiança.

4.5. Número de operações em ponto flutuante

As DNNs analisadas neste trabalho são compostas de camadas convolucionais, que executam múltiplas operações de convolução para realizar a classificação de uma dada imagem. A operação de convolução corresponde a múltiplas operações de soma e multiplicação e, portanto, operações em ponto flutuante (FLOPS). Esta seção avalia o número de FLOPS para executar a classificação de imagens. Para medir o número de FLOPS de cada modelo avaliado, executa-se o procedimento de inferência descrito na

Seção 3 e mede-se o número de FLOPs necessários até a finalização da inferência. Por exemplo, caso uma classificação seja finalizada no segundo ramo lateral, conta-se somente o número de FLOPS até esse ramo. Caso não seja possível classificá-la na borda, mede-se adicionalmente o número de FLOPS realizados até processar a camada de saída do ramo principal. Para cada nível de distorção, calcula-se o número de FLOPS médio para cada modelo avaliado. A biblioteca empregada para medir o número de FLOPS foi `pthflops 0.4.2`[5] disponível em Python.

A Figura 5 mostra o número médio de FLOPS para realizar a classificação de imagens com diferentes níveis de *blur* gaussiano. Note que a curva correspondente à DNN convencional é uma constante independentemente do nível de *blur* σ_{GB} presente na imagem, já que uma imagem é processada por todas as suas camadas neurais na DNN convencional, independentemente do nível de *blur*. Por outro lado, as curvas de EE-DNN e comitê de ramos laterais apresentam um ligeiro aumento em número de FLOPS que variam em função do nível de *blur* na imagem. Isso ocorre porque imagens com maior nível de *blur* são menos classificadas antecipadamente na borda. Consequentemente, tais imagens precisam ser processadas por mais camadas neurais, resultando em maior número de FLOPS.

A Figura 5 mostra principalmente que a utilização de soluções baseadas em DNNs com saídas antecipadas, tanto EE-DNN quanto comitê de ramos laterais, consegue reduzir drasticamente o número de FLOPS em comparação à DNN convencional, na qual todas as camadas neurais são sempre processadas. Desislavov *et al.* demonstra experimentalmente uma forte relação entre o número de FLOPS e o consumo de energia, de modo que um maior número de FLOPS indica maior número de camadas neurais, o que implica em maior consumo de energia por parte do dispositivo que executa a DNN [Desislavov et al., 2021]. Consequentemente, é possível afirmar que as EE-DNNs e o comitê de saídas antecipadas reduzem indiretamente o consumo de energia pois reduzem o número de camadas neurais processadas.

Ao comparar a EE-DNN e o comitê de ramos laterais, nota-se que o comitê realiza significativamente mais operações do que EE-DNN, em razão de classificar menos amostras antecipadamente na borda. Portanto, esse resultado indica que o comitê de ramos laterais pode consumir mais energia do que EE-DNN e necessita de um tempo maior para classificar amostras, como apresentado nos experimentos a seguir.

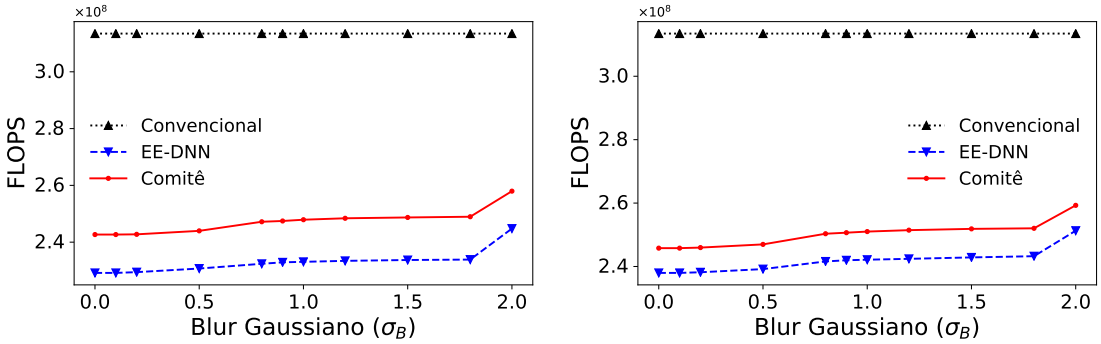
4.6. Tempo de inferência

Esta seção analisa o tempo de inferência das abordagens de DNN para diferentes níveis de *blur*. Este experimento utiliza a implementação do cenário de *offloading* adaptativo, como descrito na Seção 4.2, considerando que há três ramos laterais no dispositivo em borda. Para cada valor de nível de *blur*, calcula-se o tempo de inferência médio e seu respectivo intervalo de confiança com confiança de 95% para três abordagens avaliadas.

A Figura 6 apresenta, em função do nível de *blur* o tempo de inferência médio e seu respectivo intervalo de confiança[6] para as abordagens avaliadas. Essa figura mostra que as soluções baseadas em EE-DNNs conseguiram reduzir drasticamente o tempo de

[5] `https://pypi.org/project/pthflops/`

[6] Na Figura 6, o intervalo de confiança não está perceptível na curva

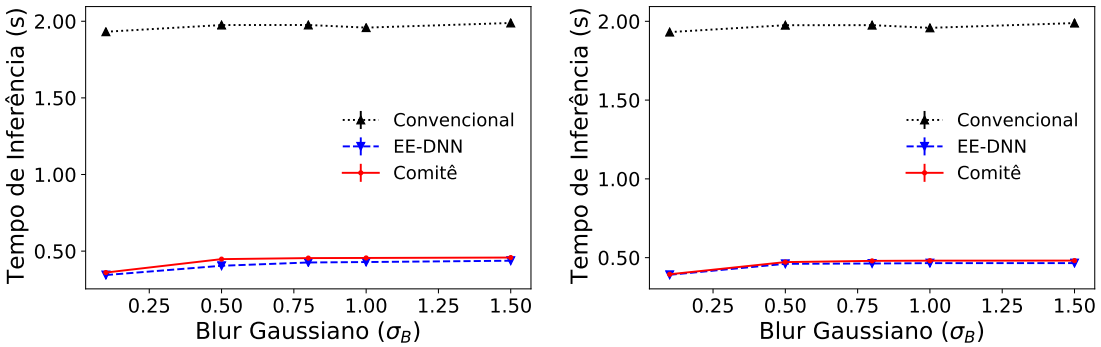


(a) $\alpha = 0.8$. (b) $\alpha = 0.9$.

Figura 5. Número de FLOPS de EE-DNNs e comitê de ramos laterais dado diferentes limiares de confiança.

inferência para classificar imagens em comparação à DNN convencional. Esse comportamento ocorre porque as soluções baseadas em EE-DNNs, tanto EE-DNN em si quanto comitê de ramos laterais, conseguem classificar um número expressivo de imagens antecipadamente na borda, o que impede a introdução de atraso de comunicação. Por outro lado, a DNN convencional sempre necessita do envio da imagem bruta à nuvem, introduzindo atraso de comunicação. Como mencionado na Seção 4.2, a nuvem é instanciada em São Paulo e, portanto, próxima do dispositivo em borda localizado no Rio de Janeiro. Mesmo assim, os resultados mostraram uma redução drástica do tempo de inferência, o que poderia ser ainda maior considerando a nuvem instanciada em regiões mais distantes.

Quando se compara a EE-DNN com comitê de ramos laterais, nota-se que os resultados do tempo de inferência são muito próximos. Esse comportamento ocorre em razão da capacidade de processamento do hardware empregado como dispositivo em borda, o NVIDIA Jetson Nano. Contudo, em outros dispositivos com menor poder de processamento, essa diferença de tempo de processamento pode ser significativa, já que o comitê necessita de um número maior de FLOPs para classificar uma imagem, como mostrado na Seção 4.5.



(a) $\alpha = 0.8$. (b) $\alpha = 0.9$.

Figura 6. Tempo de inferência médio, em segundos, da EE-DNN e comitê de ramos laterais em um cenário de *offloading* adaptativo e DNN convencional considerando cenário de processamento na nuvem.

5. Conclusão e trabalhos futuros

Este trabalhou analisou como soluções de DNN de saídas antecipadas se comportam ao receberem imagens distorcidas. Para tal, utilizou-se uma estratégia padrão de saídas antecipadas (EE-DNN) e uma estratégia de comitê de ramos laterais de uma EE-DNN. O tipo de distorção avaliado nos experimentos foi o *blur* gaussiano.

Considerando imagens sem ou com baixo nível de *blur*, este trabalho mostrou que EE-DNN e o comitê de ramos laterais obtêm um ligeiro ganho de acurácia em comparação à DNN convencional avaliada. Portanto, esses resultados sugerem a utilização de soluções baseadas em EE-DNNs em detrimento a DNN convencional para imagens sem distorção ou com baixo nível de *blur*. À medida que o nível de *blur* aumenta, a EE-DNN e o comitê de ramos laterais obtiveram acurácias próximas à DNN convencional analisada ou, em casos pontuais, ligeiramente inferior. Portanto, esses resultados mostram que as EE-DNNs e o comitê de saídas antecipadas são tão sensíveis, em termos de acurácia, quanto a DNNs convencionais. Entretanto, considerando a utilização do cenário de *offloading* adaptativo, essas estratégias baseadas em EE-DNNs conseguem oferecer diversas vantagens. Nesse cenário, a EE-DNN e o comitê de ramos laterais conseguem classificar uma parcela significativa de amostras antecipadamente na borda para todos os níveis de *blur* analisado. Consequentemente, a EE-DNN e o comitê de ramos laterais conseguem uma redução expressiva do tempo de inferência e do número de FLOPS para classificar uma imagem. Esse último indica também uma possível redução do consumo de energia. Portanto, apesar de uma ligeira redução de acurácia em casos pontuais, a utilização de EE-DNNs e comitê de ramos laterais se mostrou vantajosa, em razão da possibilidade de classificar uma quantidade significativa de imagens antecipadamente.

Como futuros passos, este trabalho visa desenvolver estratégias que tornem as EE-DNNs ainda mais robustas a imagens com distorção, considerando diferentes tipos de distorção, como ruído gaussiano e compressão da imagem. Em seguida, uma possível futura direção é desenvolver novas estratégias de comitê de ramos laterais, por exemplo, avaliando diferentes classificadores, como *random forest*.

Referências

Desislavov, R., Martínez-Plumed, F. e Hernández-Orallo, J. (2021). Compute and energy consumption trends in deep learning inference. *arXiv preprint arXiv:2109.05472*.

Dodge, S. e Karam, L. (2016). Understanding how image quality affects deep neural networks. Em *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, p. 1–6.

Dodge, S. F. e Karam, L. J. (2018). Quality robust mixtures of deep neural networks. *IEEE Transactions on Image Processing*, 27(11):5553–5562.

Fang, B., Zeng, X., Zhang, F., Xu, H. e Zhang, M. (2020). Flexdnn: Input-adaptive on-device deep learning for efficient mobile vision. Em *IEEE/ACM Symposium on Edge Computing (SEC)*, p. 84–95.

Griffin, G., Holub, A. e Perona, P. (2007). Caltech-256 object category dataset. Relatório técnico, California Institute of Technology.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. e Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Laskaridis, S., Venieris, S. I., Almeida, M., Leontiadis, I. e Lane, N. D. (2020). SPINN: synergistic progressive inference of neural networks over device and cloud. Em *International Conference on Mobile Computing and Networking (MobiCom)*, p. 1–15.

Li, E., Zeng, L., Zhou, Z. e Chen, X. (2019). Edge ai: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications*, 19(1):447–457.

Liu, Z., Lan, G., Stojkovic, J., Zhang, Y., Joe-Wong, C. e Gorlatova, M. (2020). Collabar: Edge-assisted collaborative image recognition for mobile augmented reality. Em *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, p. 301–312.

Pacheco, R. e Couto, R. (2020). Introduzindo a qualidade de imagem como uma nova condição de particionamento de dnn na borda. Em *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 43–56, Porto Alegre, RS, Brasil. SBC.

Pacheco, R., Couto, R. e Simeone, O. (2021a). Calibration-aided edge inference offloading via adaptive model partitioning of deep neural networks. Em *IEEE Int. Conf. Communications (ICC)*, p. 1–6.

Pacheco, R. G., Bochie, K., Gilbert, M. S., Couto, R. S. e Campista, M. E. M. (2021b). Towards edge computing using early-exit convolutional neural networks. *Information*, 12(10):431.

Pacheco, R. G., Oliveira, F. D. e Couto, R. S. (2021c). Early-exit deep neural networks for distorted images: providing an efficient edge offloading. Em *IEEE Global Communications Conference (GLOBECOM)*, p. 1–6.

Qendro, L., Campbell, A., Lio, P. e Mascolo, C. (2021). Early exit ensembles for uncertainty quantification. Em Roy, S., Pfohl, S., Rocheteau, E., Tadesse, G. A., Oala, L., Falck, F., Zhou, Y., Shen, L., Zamzmi, G., Mugambi, P., Zirikly, A., McDermott, M. B. A. e Alsentzer, E., editors, *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, p. 181–195. PMLR.

Rozsa, A., Gunther, M. e Boult, T. E. (2016). Towards robust deep neural networks with bang. *arXiv preprint arXiv:1612.00138*.

Secci, F. e Ceccarelli, A. (2020). On failures of RGB cameras and their effects in autonomous driving applications. Em *IEEE International Symposium on Software Reliability Engineering (ISSRE)*, p. 13–24.

Teerapittayanon, S., McDanel, B. e Kung, H.-T. (2016). Branchynet: Fast inference via early exiting from deep neural networks. Em *IEEE International Conference on Pattern Recognition (ICPR)*, p. 2464–2469.

Wang, M., Mo, J., Lin, J., Wang, Z. e Du, L. (2019). Dynexit: A dynamic early-exit strategy for deep residual networks. Em *IEEE International Workshop on Signal Processing Systems (SiPS)*, p. 178–183.