A Light Implementation of a 3D Convolutional Network for Online Gesture Recognition

F. Brandolt, and F. Vargas, *Senior Member, IEEE*

***Abstract*—With the advancement of machine learning techniques and the increased accessibility to computing power, Artificial Neural Networks (ANNs) have achieved state-of-the-art results in image classification and, most recently, in video classification. The possibility of gesture recognition from a video source enables a more natural non-contact human-machine interaction, immersion when interacting in virtual reality environments and can even lead to sign language translation in the near future. However, the techniques utilized in video classification are usually computationally expensive, being prohibitive to conventional hardware. This work aims to study and analyze the applicability of continuous online gesture recognition techniques for embedded systems. This goal is achieved by proposing a new model based on 2D and 3D CNNs able to perform online gesture recognition, i.e. yielding a label while the video frames are still being processed, in a predictive manner, before having access to future frames of the video. This technique is of paramount interest to applications in which the video is being acquired concomitantly to the classification process and the issuing of the labels has a strict deadline. The proposed model was tested against three representative gesture datasets found in the literature. The obtained results suggest the proposed technique improves the state-of-the-art by yielding a quick gesture recognition process while presenting a high accuracy, which is fundamental for the applicability of embedded systems.**

***Index Terms*—Gesture Recognition, Online Classification, 3DCNN.**

I. Introdução

DESDE a criação dos computadores, pesquisadores acadêmicos e da indústria vêm buscando maneiras mais intuitivas para nós interagirmos com esses dispositivos[1]. Além dos meios convencionais de interação com computadores como mouse, teclado e telas sensíveis ao toque, outros meios, como por exemplo, através da fala (ex.: Alexa e Google Assistant) e de gestos (ex.: realidade virtual) vêm se tornando mais comuns, por serem maneiras mais naturais de comunicação para seres humanos[2]. O reconhecimento de gestos em especial pode ser útil em diversas áreas, como uma forma de controlar equipamentos de forma remota [3], uma forma imersiva de interação durante experiências de realidade virtual e até mesmo na transcrição de linguagem de sinais.

Técnicas de reconhecimento de gestos podem ser divididas em dois principais grupos: técnicas de contato e técnicas baseadas em visão [2]. O primeiro grupo trata de técnicas que envolvem o uso de dispositivos adicionais, que devem ser segurados (por exemplo um controle) ou vestidos pelo usuário. Os equipamentos vestíveis, posicionados nos braços e/ou mãos do usuário tem como exemplo mais comum luvas. Através de dispositivos capazes de estimar posição e movimento, como acelerômetros e giroscópios, esses dispositivos conseguem mapear o posicionamento dos membros do usuário, de forma a extrair a informação de sua movimentação.

O segundo grupo, técnicas baseadas em visão, utiliza-se de câmeras, que capturam o movimento do usuário, sem a necessidade deste estar utilizando equipamento adicional. Além da captura mais tradicional de quadros RGB (imagem em três canais de cores), as capturas podem conter imagens de profundidade, sinais infravermelhos e outras formas de captura de imagem. Nessa metodologia, a informação da movimentação do usuário deverá ser extraída através de técnicas de visão computacional a partir da captura de tais sensores, que não estão diretamente conectados ao usuário. Técnicas baseadas em visão são consideradas menos intrusivas ao usuário [2], permitindo a utilização de forma mais ampla.

Com o avanço no poder computacional de dispositivos eletrônicos, técnicas de aprendizado de máquina ganharam popularidade tanto no meio acadêmico quanto na indústria para solucionar problemas de processamento de linguagem natural [1] (por exemplo, reconhecimento de voz [4]) e visão computacional (por exemplo, leitura de placas de carro a partir de câmera de vídeo). Notoriamente, o desenvolvimento de Redes Neurais Convolucionais (do inglês, *Convolutional Neural Network*, ou CNN) compõe o estado da arte em classificação de imagens[5], [6] e mais recentemente vídeos. Um dos grandes desafios para se utilizar dessas técnicas de classificação de vídeo para solucionar a tarefa de reconhecimento de gestos em aplicações embarcadas é a complexidade computacional das mesmas. Isso se deve não só ao volume de dados presente em um vídeo, mas também devido às complexas técnicas utilizadas no processamento dos dados utilizando-se de redes neurais, como redes de convolução 3D (3DCNN), redes LSTM (Long-Short Term Memory), processamento de *Optical Flow* (método de extração de quadros que representa movimento a partir de quadros do vídeo), etc.

Enquanto grande parte dos trabalhos publicados na área de classificação de vídeos tem como foco a obtenção de modelos que maximizam a acurácia [7], a otimização dessas técnicas para o uso em sistemas embarcados, é pouco explorada. Funções de reconhecimento de gestos podem ter diversas aplicações quando presentes em um sistema embarcado, porém o alto custo computacional da grande maioria das técnicas que representam o estado-da-arte da área impossibilita a implementação dessas em sistemas com poder de processa-

Fabian Vargas is full professor of the Electrical Engineering Department of the Catholic University - PUCRS, Brazil. He is a Golden Core Member of the IEEE Computer Society since 2003.

Fabio Brandolt Baldissera (fb.baldissera@gmail.com) obtained his MSc degree from the Catholic University - PUCRS on Oct 2019.

mento mais modesto.

Este artigo propõe um modelo de rede neural convolucional de reconhecimento de gestos projetado para o uso em sistemas embarcados. Mais especificamente, reconhecimento *online* de gestos, isto é, fazendo predições para cada novo quadro de imagem obtido, utilizando-se apenas de captura de vídeo em RGB a fim de não requerer uma câmera especial (que capture profundidade, por exemplo) para o uso. Exemplos de hardwares para executar a técnica proposta incluem Raspberry Pi 4 ou smartphones modernos (contendo SoC, *System on Chip*, Snapdragon atual, por exemplo). Outras análises necessárias para uma aplicação em tempo real como a aquisição dos quadros e clareza das imagens obtidas (iluminação, constraste, etc.) não são abordadas neste artigo a fim de focar na implementação da rede neural em si. Esta técnica foi desenvolvida baseada em modelos de alta acurácia visando aplicações genéricas, observando-se quais estruturas seriam viáveis para implementar uma versão própria para sistemas embarcados, tendo em vista a minimização da complexidade computacional e mantendo acurácias compatíveis com essas técnicas.

II. Estado-da-Arte

O recente desenvolvimento de redes convolucionais (CNNs), especialmente arquiteturas ”profundas”, isto é, contendo várias camadas de convolução, demonstrou a grande capacidade dessas redes em tarefas de visão computacional. Em bancos de dados de imagens como ImageNet e CIFAR-10, redes que são majoritariamente compostas por camadas de convolução como ResNet, DenseNet e Inception [8] destacam-se devido à alta capacidade de diferenciar classes de imagens. Com o sucesso obtido para visão computacional em imagens, a área de classificação de vídeo começa a ganhar mais atenção, juntamente com o desenvolvimento de bancos de dados voltados à classificação de vídeos e o surgimento de hardware mais adequado para tal tarefa.

A classificação de vídeos contém novos desafios em relação a classificação de imagens. Com o aumento da quantidade de dados a serem processados (várias imagens para representar um vídeo) e a necessidade de não só ser capaz de extrair características espaciais dos quadros (relações de pixeis com seus vizinhos) como também ser capaz de interpretar características temporais (capacidade de relacionar características entre quadros vizinhos), a tarefa de classificação de vídeo pode ser considerada de alta complexidade. Além disso, enquanto que, por exemplo, para a diferenciação de vídeos de esportes (como no conjunto de dados Sports-1M) um quadro estático pode representar informação suficientemente relevante para a classificação do vídeo como um todo, devido a presença de elementos visuais característicos de uma certa classe (por exemplo, uma cesta em um jogo de basquete), para a classificação de gestos os elementos visuais de classes diferentes são semelhantes entre si (presença de mãos, braços, dedos, rostos, etc.). É comumente necessário que sejam analisados múltiplos quadros e necessariamente em sequência para que o conceito de um gesto dinâmico (em que a movimentação do usuário é parte do gesto) seja diferenciado de um similar.

A seguir, serão apresentadas algumas das técnicas que são comummente utilizadas em reconhecimento de gestos e classificação de vídeos em geral.

A. Redes Convolucionais + Redes LSTM

Como dito anteriormente, CNNs tem destaque na classificação de imagens, sendo capazes de extrair a informação espacial de quadros individuais, com uma acurácia suficientemente alta para uma grande quantidade de aplicações.

Aliado a isso, redes LSTM obtiveram resultados que compõe o estado-da-arte em tarefas de análise de dados sequenciais, como reconhecimento de voz e análise de sentimento em textos [1]. Combinando a capacidade dessas duas estruturas em uma rede neural, de forma utilizar uma CNN para extração espacial e uma rede LSTM para a extração temporal de informações, trabalhos como [9] refletem o estado da arte dessa abordagem na classificação de vídeos. Uma das características que possibilita a experimentação e o uso de CNNs de forma mais acessível é um processo chamado ”transferência de aprendizagem” (do inglês, *transfer learning*). Através dele é possível dar nova funcionalidade a uma CNN previamente treinada para outro uso. Assim, a reutilização de CNNs bem sucedidas em bancos de dados de imagens como ImageNet, permite auxiliar o treinamento dessas novas redes voltadas para classificação de vídeos, permitindo a reutilização dos filtros já treinados anteriormente.

Um exemplo de técnica que usa este tipo de estrutura é demonstrada em [9]. Neste artigo, os autores criam uma rede baseada em atenção, que avalia em quais partes do vídeo deve-se analisar mais profundamente. Os autores usam quadros RGB e *flow frames* (derivados de um processo chamado *Optical Flow* que cria quadros que representam visualmente o movimento entre os quadros) como a entrada de suas redes, em um formato ”two-stream” (baseado em uma técnica com mesmo nome, em que quadros RGB e *flow frames* são processados separadamente e concatenados em uma camada final responsável pela classificação em si) para bancos de reconhecimento de atividades.

B. Redes Convolucionais 3D

A adaptação das redes de convolução 2D para operar em uma dimensionalidade a mais é a rede convolucional 3D (3DCNN). Quadros de vídeo são imagens com representação bidimensionais. Quando vários quadros de vídeo são combinados, a informação passa a ser representada por três dimensões: altura dos quadros, largura dos quadros e os diferentes quadros em si. Essas redes são capazes de extrair características espaço-temporais simultaneamente, devido a seus filtros conterem uma dimensionalidade a mais. Isto é, cada filtro afeta vários quadros de vídeo, podendo desenvolver captar informações sequenciais presentes entre esses. Esse tipo de rede se mostra presente em diversas técnicas do estado-da-arte de classificação de vídeo, e tem demonstrado-se efetivo em tarefas como: geração de legendas para descrever ações em vídeos [10], reconhecimento de objetos em tempo real [11], classificação de ações [9], reconhecimento de gestos

(utilizando sensores de profundidade) [12] e classificação de vídeos de modo geral [9]. Redes de convolução 3D são possivelmente as mais comuns em trabalhos atuais de classificação de vídeo

C. Aplicabilidade em Sistemas Embarcados

A grande dificuldade encontrada para a utilização dessas redes citadas é que naturalmente a tarefa de reconhecimento de gestos demanda bastante poder computacional. Por esta razão, grande parte dos trabalhos publicados na área em questão tem como foco a predição *offline*, ou seja, após a recepção por completo da amostra, a rede atribui a esta uma etiqueta. Por outro lado, a classificação *online* consiste em ser capaz de classificar a amostra quadro a quadro, identificando, por exemplo, quando o determinado gesto foi executado. A classificação *online* é o modo de operação necessário para que aplicações de tempo real reconheçam um determinado gesto concomitantemente com a sua realização. Existem outros aspectos a serem observados na classificação *online*, como o delay de classificação (quantos quadros são necessários para identificar um gesto), a sensibilidade da rede (para compensar a quantidade de falsos positivos ou falsos negativos) e a capacidade de identificar quando um gesto terminou para evitar computá-lo múltiplas vezes.

Mesmo em trabalhos que abordam a classificação *online* de gestos, nem sempre existe uma preocupação com a complexidade computacional da técnica, ou seja, uma preocupação de fazê-la de forma eficiente, com recursos de *hardware* o mais simples possíveis. Por consequência a vasta maioria das técnicas tem por objetivo principal atingir a maior acurácia possível, sem se peocupar necessariamente com o custo que a obtenção deste objetivo implica em termos de implementação em hardware da rede para operar em sistemas embarcados.

Tendo isso em consideração, este artigo apresenta um modelo de rede neural convolucional voltado para o reconhecimento *online* de gestos em sistemas embarcados. Dito isto, entende-se que esta é a maior contribuição deste trabalho: desenvolver uma técnica dedicada para o reconhecimento *online* de gestos que ao mesmo tempo privilegia estruturas de *hardware* de baixa complexidade de forma a facilitar a implementação em sistemas embarcados. A rede deve ser capaz de operar continuamente, identificando quando um gesto foi executado (e apenas uma vez por gesto) com um atraso compatível com outras técnicas de classificação *online*. Optou-se também por não fazer uso de outros tipos de sensores (como por exemplo infravermelho, profundidade, etc.), para tornar a utilização da técnica proposta o mais simples possível.

III. Técnica Proposta: Real-Time 3D 16 Frames (RT3D_16F)

Para o desenvolvimento da rede proposta, as estruturas de rede que compõe o estado da arte foram analisadas. Como grande parte dessas redes não tem como objetivo a execução em sistemas embarcados, certas estruturas que compõe essas redes acabam não sendo boas candidatas para tal fim. Um exemplo disso é o processamento de *flow frames* utilizando algoritmos de *Optical Flow* [13], que está amplamente presente em diversas técnicas para reconhecimento de gestos. Em testes preliminares, o uso desse tipo de processamento se mostrou computacionalmente caro (elevado tempo de computação), mesmo em versões mais simplificadas do algoritmo.

Uma estrutura presente na maioria das técnicas do estado-da-arte em reconhecimento de gestos é a convolução 3D. E é nesta que se baseiam os modelos propostos neste artigo. O nível de complexidade computacional de uma camada de uma rede 3DCNN depende da dimensionalidade do tensor de entrada. Um tensor, neste contexto, serve como uma generalização de representações numéricas dimensionais, como por exemplo vetores (1D) e matrizes (2D), porém podendo representar qualquer representação contendo N dimensões. A fim de reduzir a complexidade de processamento dos quadros de vídeo diretamente em uma rede 3DCNN, o modelo proposto possui uma rede 2DCNN que recebe os quadros e pré-processa estes, reduzindo assim a dimensionalidade dos dados, além de auxiliar na extração de características espaciais (ser capaz de identificar elementos presentes nos quadros, como mãos, braços, etc.).

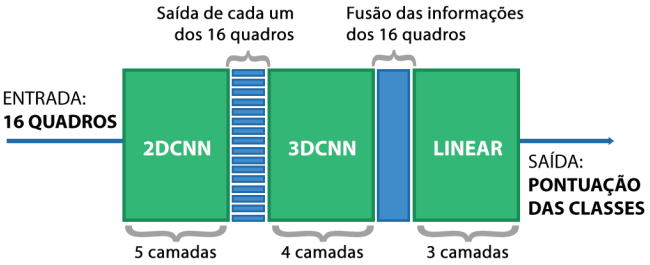

Fig. 1. Diagrama de blocos alto nível do modelo proposto.

A Fig. 1 mostra uma visão geral da estrutura da rede proposta. Ela é composta por uma 2DCNN, responsável por pré-processar os quadros, uma 3DCNN que é a principal forma de extração das características espaço-temporais da rede e uma série de camadas lineares que se utilizam das características extraídas da rede para gerar a pontuação de cada classe de gesto. Entre as camadas em verde, é mostrada a representação da informação dos quadros em azul. Após a 2DCNN, a informação de cada quadro ainda é individual, enquanto que após as camadas da 3DCNN, a informação de todos os quadros é fundida. Este detalhe terá implicações que serão discutidas posteriormente na Seção III-A.

Além disso, outras duas medidas foram levadas em consideração na implementação da rede. A primeira se trata da utilização de uma baixa taxa de aquisição de quadros, ou seja, nem todos os quadros de vídeo oferecidos pelos bancos de dados utilizados são inseridos na rede para fazer o reconhecimento do gesto. Nos bancos de dados Jester e nvGesture, os resultados foram obtidos com taxas de aquisição de 3 e 6 FPS (quadros por segundo, do inglês, *Frames Per Second*), respectivamente. Isto tem uma implicação relevante quanto à performance da rede. Como na classificação *online* as predições são feitas a cada novo quadro adquirido, considerando um mesmo hardware alvo, uma taxa de aquisição de quadros menor implica em mais tempo para que cada operação da rede seja processada.

TABELA I
CAMADAS DE CONVOLUÇÃO 2D E 3D PARA O MODELO PROPOSTO

Camada	Canais Entrada	Canais Saída	Filtro	*Pooling*
2D Conv	3	16	[3,3]	Não
2D Conv	16	32	[3,3]	Não
2D Conv	32	64	[3,3]	[2,2]
2D Conv	64	128	[3,3]	[2,2]
2D Conv	128	256	[3,3]	[2,2]
3D Conv	256	256	[3,3,3]	Não
3D Conv	256	256	[3,3,3]	[2,2,2]
3D Conv	256	256	[3,3,3]	[2,1,1]
3D Conv	256	256	[3,3,3]	[2,2,2]

A outra medida levada em consideração na rede é a reutilização de informações de uma janela de quadros para outra. No modo de operação *online*, existe uma janela de quadros sendo observada em um dado momento, contendo N quadros. Ao adquirir um quadro novo, esta janela perde o quadro mais antigo e adiciona o quadro recém adquirido na janela. Assim, entre uma janela de operação e a próxima existem $N-1$ quadros em comum. Fazendo o uso de estruturas de rede que não mesclam dados de múltiplos quadros, isto é, computações matemáticas que não envolvam mais de um quadro para seu resultado, é possível reutilizar os dados processados dessa rede para o próximo ciclo de predição. No caso da 2DCNN presente, a computação de cada quadro é feita de forma individual para cada quadro, ou seja, como existem $N-1$ quadros em comum com a janela do ciclo anterior, ao invés de processar novamente os resultados da 2DCNN para os quadros presentes em ambas as janelas, é possível reutilizar a informação do ciclo anterior, restando apenas o processamento da 2DCNN para o quadro novo na janela. O bloco 3DCNN presente não permite fazer uso desta redundância de dados, pois possui filtros multidimensionais que fundem os dados de todos os quadros de entrada, ou seja, não é possível separar o tensor de forma a isolar qual parte resultantes da rede foi influênciada por apenas um determinado quadro. A execução deste processo na rede proposta reduziu em mais de 60% o tempo de execução *offline* da rede.

As camadas que compõe os blocos de 2DCNN e 3DCNN estão listadas na Tabela I. Nesta, estão descritos os principais parâmetros da rede. Além disso existem camadas de ativação, utilizando ReLU [14] (função de ativação que mantém o sinal idêntico para valores positivos e zero para valores negativos) e camadas de normalização (*batch normalization*), que foram omitidas por simplicidade. O bloco linear da rede consiste de 3 camadas. Este bloco tem a função de reduzir o tensor de saída da rede 3DCNN a fim de transformar progressivamente os 6144 nós em 3072 nós, em seguida em 1024 nós, e finalmente na quantidade de classes existentes no modelo, dependendo do conjunto de dados.

A. Operação Online

As redes são treinadas de forma *offline*, ou seja, recebendo trechos de gestos apenas e atribuindo uma classe para os trechos mostrados. Na operação *online* é necessário adaptar o funcionamento da rede para que seja possível operar de forma contínua, identificando quando um gesto foi realizado e só então atribuir uma etiqueta para a previsão. Para isso é utilizada uma janela dos 16 quadros mais recentes obtidos, que se atualiza a cada novo quadro adquirido. O modelo proposto avalia a janela em questão e salva o softmax desse resultado para uma buffer contendo as N previsões mais recentes. Caso a média de alguma classe nesse buffer supere um limite de confiança de rede (C_{th}), o algoritmo então identificará que este gesto ocorreu. Para evitar que um mesmo gesto que o usuário tenha realizado acuse várias vezes a presença de um gesto na rede (indicando erroneamente que este foi feito mais de uma vez), o algoritmo após atribuir uma etiqueta tem um tempo de *cooldown*, em que novas etiquetas não serão atribuídas, mesmo se alguma classe ultrapasse o limite de confiança. Isso faz com que a a janela que armazena os quadros mais recentes se renove, e evitando que a mesma sequência de quadros que gerou uma classificação de um gesto, o faça novamente. A Fig. 2 mostra um fluxograma de operação da técnica em operação *online*.

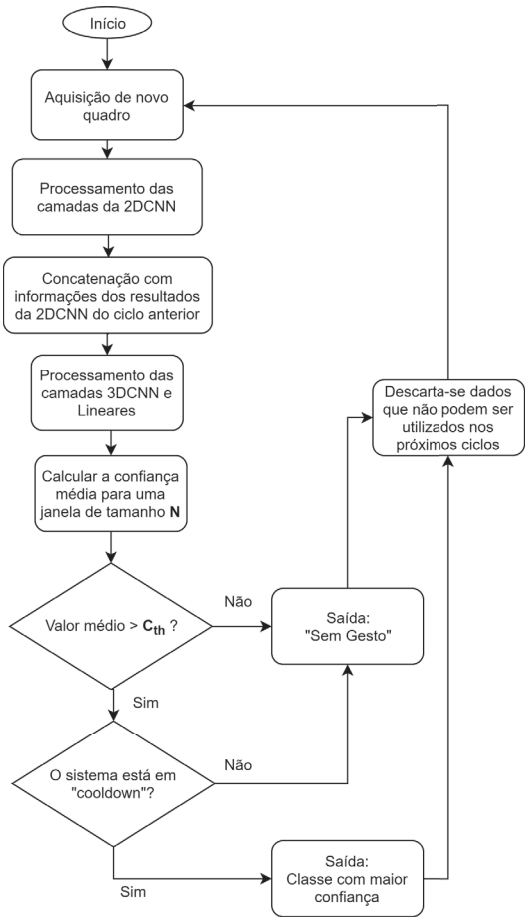

Fig. 2. Fluxograma de operação da rede em operação *online*.

B. Treinamento

Neste artigo, foram utilizados 3 bancos de dados contendo vídeos de gestos: Jester [15], nvGesture [11] e EgoGesture [16]. Estes serão utilizados para comparações da rede em operações *offline* e *online*. Nas camadas lineares da rede foram

inseridas camadas de *dropout*, com probabilidade de 60%. A rede foi incialmente treinada no conjunto de dados Jester, e esta versão foi utilizada como base para o treinamento nos bancos de dados nvGesture e EgoGesture. Esta medida foi utilizada pois o conjunto de dados com maior número de amostras é o Jester, e o treinamento via *transfer learning* poderia evitar *overfitting* no restante dos bancos, que possuem uma quantidade menor de amostras. Para o banco nvGesture as únicas camadas modificadas foram as lineares, visto que overfitting foi observado ao mudar os parâmetros dos blocos 2DCNN e 3DCNN. Já o conjunto de dados EgoGesture foi treinado inicialmente apenas modificando as camadas lineares e posteriormente permitindo que os parâmetros da 2DCNN e 3DCNN fossem optimizados, visto que neste caso um ganho real de performance na acurácia foi observado.

Em suma, destacamos os três principais recursos inovadores propostos neste trabalho. Primeiro, a inserção de uma rede 2DCNN para pré-processar os quadros antes do processamento principal de extração das características espaço-temporais realizado pela rede 3DCNN. Isso permite a redução da dimensionalidade dos dados ao ser inserida na 3DCNN, além de permitir a reutilização da informação entre ciclos, reduzindo o tempo necessário para a classificação *online*. O segundo recurso foi a utilização de uma baixa taxa de aquisição de quadros a fim de reduzir a quantidade de dados a ser analisado e disponibilizar mais tempo entre quadros para a execução da computação do modelo proposto. O último recurso destacado aqui é o sistema de adaptação de um modelo *offline* para a classificação *online*. O modelo proposto é de simples implementação, apresentando baixo custo adicional de performance e utiliza um sistema de *cooldown* para evitar a múltipla computação de um mesmo gesto.

IV. Resultados

Nesta seção serão analisados os resultados obtidos com o modelo proposto utilizando os bancos de dados Jester [15], nvGesture [11] e EgoGesture [16] para a avaliação *offline* e comparação com outras técnicas do estado da arte, e apenas os bancos nvGesture e EgoGesture para a avaliação *online* da técnica.

A. Resultados Offline Conjunto de Dados Jester

Este conjunto de dados consiste de 148.092 amostras de vídeos contendo gestos divididos em 27 classes (incluindo ”Nenhum Gesto” e ”Fazendo outras coisas”). Os vídeos são compostos de quadros capturados a 12 quadros por segundo, sendo levemente variável o número de quadros por amostra, mas de forma geral esta contém apenas a execução do gesto em si. Cabe ressaltar que quanto maior o número de quadros por segundo, maior é a informação de movimento contida nas imagens e, portanto, maior é a expectativa de uma acurácia mais elevada para a técnica. A visão da câmera é frontal, típica de webcams de notebooks. O tamanho dos quadros têm altura (H) fixa de 100 pixels e largura (L) variável entre 100 e 200 pixeis, que para uso na rede são todos recortados no formato $[L, H] = [140, 100]$.

O conjunto de dados é dividido em 3 partes: treinamento, validação e teste, divididos em uma proporção 8:1:1. O último grupo não contém as etiquetas das amostras, pois é utilizado para um *leaderboard* (lista ordenada por acurácia das técnicas que submeteram seus resultados) no site da empresa criadora do banco: twentybn.

TABELA II
Acurácia do Conjunto de Dados Jester

Técnica	Acurácia
MFNet[17]	96,22
Motion Fused Frames (MFF) [18]	96,28
RT3D-16F	92,65
20BN Jester System [15]	82,34

A Tabela II apresenta os valores obtidos no subgrupo de testes do conjunto de dados Jester. Mesmo utilizando apenas metade dos quadros de vídeo disponíveis, foi possível alcançar uma acurácia acima de 90%. Embora não venha a competir diretamente com outras técnicas focadas em classificação *offline*, a reduzida complexidade computacional da técnica auxilia na utilização desta em sistemas embarcados. Como as amostras desse conjunto de dados são inapropriadamente curtas para simular a operação *online* contínua do modelo, este banco será utilizado apenas para a comparação *offline* de resultados e para o treino dos modelos dos demais bancos de dados.

B. Resultados Offline Conjunto de dados nvGesture

Este conjunto de dados contém 1.532 amostras de vídeo de 25 classes de gestos distintas. Este conjunto de dados foi capturado simulando a utilização em um carro, sendo o ponto de vista da câmera posicionado no painel central do carro, observado o usuário de frente. O conjunto de dados é dividido em treinamento e validação, numa proporção de 7:3. A taxa de aquisição de quadros deste conjunto de dados é de 30 quadros por segundo, e os quadros têm tamanho fixo de [320, 240] pixels. As amostras de vídeo são compostas por longos trechos de simulação do usuário dirigindo (mãos ao volante, olhando para a parte anterior do carro) contendo um gesto ao decorrer do vídeo. Além das imagens RGB, o conjunto de dados oferece outras capturas (como profundidade e infravermelho) que não serão utilizadas neste artigo.

TABELA III
Acurácias Obtidas no Conjunto de Dados nvGesture

Técnica	Tamanho da janela	Acurácia
RT3D-16F	16 quadros	**67,42**
C3D [7]	16 quadros	62,67
ResNeXt-101 [7]	16 quadros	66,40
ResNeXt-101 [7]	32 quadros	78,63
R3DCNN [11]	32 quadros	74,10

A Tabela III mostra os resultados obtidos no conjunto de dados nvGesture, comparado a outras técnicas fazendo o uso apenas dos quadros RGB. Este conjunto de dados, apesar de ter um número de classes similar ao Jester (25

contra 27 classes), possui resultados de acurácia publicados significativamente menores, possivelmente devido ao reduzido número de amostras por classe de gestos (1.532 amostras totais contra 148.092 amostras totais no Jester). Quando comparada a outras técnicas que utilizam o mesmo número de quadros como entrada da rede, a abordagem proposta possui uma acurácia comparável.

C. Resultados Offline Conjunto de Dados EgoGesture

Este conjunto de dados possui uma visão diferente de ambos os anteriores, num ponto de vista em primeira pessoa (capturado com uma câmera acima da cabeça do usuário). O banco possui 2.081 amostras, cada uma contendo trechos que contém de 9 a 14 gestos, totalizando 24.161 amostras de gestos individuais. O banco é dividido em grupos para treinamento, validação e teste numa proporção de 3:1:1. O banco possui 83 classes de gestos distintas, capturado a 30 quadros por segundo e num tamanho de [640, 480]. Além disso o conjunto de dados ainda conta com uma captura de profundidade, que não é utilizada neste artigo.

TABELA IV
ACURÁCIAS OBTIDAS NO GRUPO DE VALIDAÇÃO DO BANCO EGOGESTURE - GESTOS ISOLADOS

Técnica	Tamanho da janela	Acurácia
RT3D-16F-WIDE	16 quadros	**86,09**
VGG-16 [16]	16 quadros	62,50
VGG-16 + LSTM [16]	16 quadros	74,70
ResNeXt-101 [7]	16 quadros	90,94
C3D+LSTM+RSTTM [16]	16 quadros	89,30
ResNeXt-101 [7]	32 quadros	93,75

A Tabela IV mostra os resultados obtidos na classificação *offline* do banco EgoGesture. Os gestos para esse teste foram analisados isoladamente, visto que cada amostra tem múltiplos gestos. Novamente, os resultados obtidos nesse conjunto de dados são próximos aos de outras técnicas que utilizam o mesmo número de quadros como entrada de suas redes, porém consideravelmente abaixo de técnicas que utilizam mais quadros para a representação do gesto.

D. Resultados Online - Conjunto de Dados nvGesture

Para a computação da acurácia das predições *online*, a etiqueta correta de cada amostra é uma sequência de gestos (em oposição a apenas um gesto). Para uma dada amostra, a acurácia é computada utilizando a distância de Levenshtein[7] entre as duas sequências (prevista pela rede e a verdadeira), e a acurácia é computada como sendo um (1) subtraído da distância de Levenshtein dividida pelo número de elementos na sequência verdadeira, como proposto em [7].

Embora este conjunto de dados não possua amostras com mais de um gesto (para formar uma sequência) o uso da distância Levenshtein tem efeito igual a verificar se a rede foi capaz de acertar o gesto contido na amostra. A predição pode conter nenhum ou múltiplos elementos (dependendo da sensibilidade da rede), que de ambas as formas resultariam em uma acurácia nula para aquela amostra.

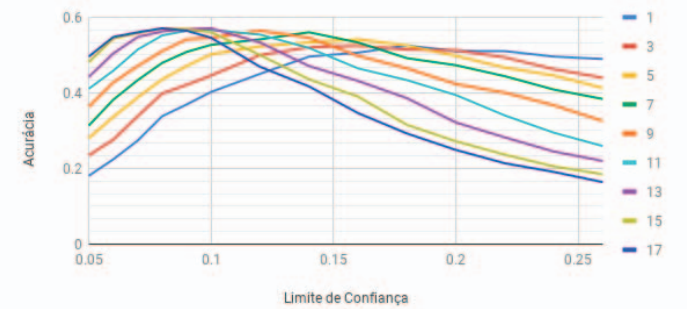

Fig. 3. Resultados de acurácias obitdas para o conjunto de dados nvGesture. Eixo x representa o impacto da variação do limite de confiança C_{th}, e cada linha representa o número N de predições cuja média é comparada.

Os testes *online* foram realizados para um intervalo de limite de confiança da rede (C_{th}) e um intervalo do número de resultados amostrados para verificar a média (N). Esses valores foram escolhidos via experimentação para encontrar as melhores combinações de acurácia (utilizando a distância Levenshtein) e o atraso encontrado para cada predição (medido em número de quadros antes de terminar o gesto em que a rede identificou o mesmo). A Figura 3 mostra a acurácia obtida nos testes obtidos. É possível notar que dependendo do N a ser analisado, o nível de confiança da rede ótimo é diferente. Utilizando-se um N maior foi possível obter resultados melhores de acurácia de forma geral.

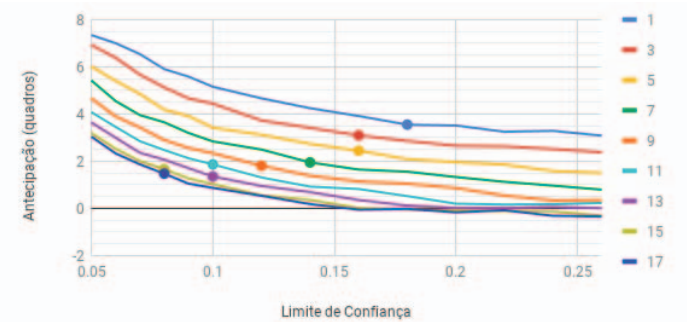

Fig. 4. Resultados do atraso para o conjunto de dados nvGesture. Eixo x representa o impacto da variação do limite de confiança C_{th}, e cada linha representa o número N de predições cuja média é comparada. Pontos marcados na linha representam o ponto de máxima acurácia para o mesmo teste.

A Figura 4 mostra o atraso observado para a predição do gesto, medido em número de quadros antes do término da performance do gesto, assim um valor maior implica que foi possível detectar o gesto de forma antecipada. Os pontos marcados nas linhas representam a maior acurácia obtida no gráfico anterior. É possível perceber que existe uma troca entre acurácia e delay, é possível, dependendo da necessidade da aplicação, sacrificar acurácia para ganhar um pouco em tempo de resposta da rede.

De forma geral, o melhor resultado de acurácia obtido na rede foi de 57,05% e uma antecipação da predição média de 1.66 quadros (276ms usando uma captura a 6 FPS), utilizando $C_{th} = 0.08$ e $N = 15$. O mesmo trabalho que propõe a métrica utilizando a distância de Levenshtein [7] conseguiu obter uma acurácia de 77,39% no mesmo conjunto de dados,

no entanto utilizando quadros de profundidade oferecidos pelo conjunto de dados. Não foram disponibilizados resultados referentes a apenas o uso de quadros RGB.

E. Resultados Online - Conjunto de Dados EgoGesture

O mesmo procedimento de testes foi adotado para o banco EgoGesture.

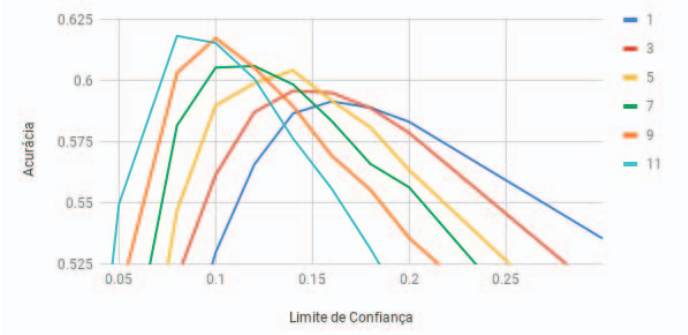

Fig. 5. Resultados de acurácias obtidos para o conjunto de dados EgoGesture. Eixo x representa o impacto da variação do limite de confiança C_{th}, e cada linha representa o número N de predições cuja média é comparada.

A Figura 5 mostra o resultado da acurácia obtido para o conjunto de dados EgoGesture. Um comportamento similar observado ao anterior é observado, em que há um ponto ótimo de confiança (C_{th}) para cada N. Apesar deste conjunto de dados ter uma acurácia maior nos testes *offline*, as amostras utilizadas para o teste *online* são mais longas e contém múltiplos gestos, o que torna a tarefa relativamente mais complexa e mais próxima daquela esperada em uma aplicação real. A antecipação observada nos pontos de maior acurácia para cada N resultou numa média de três quadros (de 1,3 até 5,6, dependendo do ponto analisado).

Além disso, o efeito do tempo de *cooldown* foi observado para intervalos de 4 à 64 quadros. Foi possível perceber que tanto um intervalo longo ou curto implica em perda de acurácia da rede, sendo o tempo ótimo, nestes testes, o suficiente para mudar todos os quadros da janela (16 quadros, ou 2.66 segundos). Um período curto pode resultar na computação de um gesto múltiplas vezes, enquanto um intervalo longo não permite gestos consecutivos em um curto intervalo de tempo.

F. Tempo de Computação

Para uma avaliação da complexidade computacional da técnica, foi elaborado um teste para verificar o tempo necessário para executar 1000 janelas de quadros pela rede. O modelo proposto foi comparado com a técnica ResNext-101 [7], previamente adaptados para o banco EgoGesture. Os testes foram executados em um computador utilizando uma placa de vídeo GTX 1080 TI (11 GB VRAM) e um processador i7 6700k. Apesar do hardware citado não ser típico de sistemas embarcados, para fins de comparação, foi optado utilizar um hardware semelhante ao utilizado pelos autores da técnica comparada. É importante ressaltar que dependendo do hardware utilizado, uma técnica ou outra pode ser beneficiada. Assim sendo, este teste tem o simples propósito de ilustrar uma análise comparativa , e ressaltar a diferença de complexidade das técnicas.

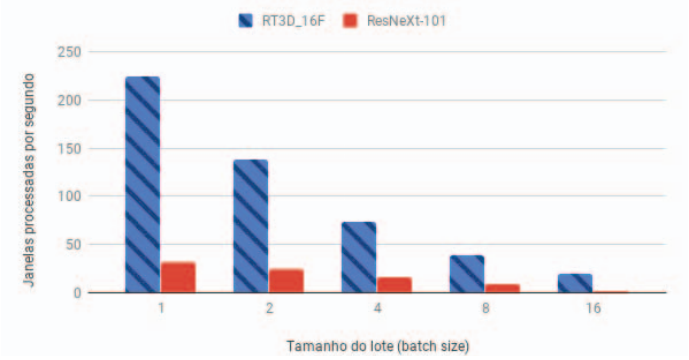

Fig. 6. Classificações por segundo - modelos utilizados para o conjunto de dados EgoGesture.

A Figura 6 mostra os números de execuções de janelas de quadros (ou seja, o tempo de classificação *offline*) por segundo obtidos pela técnica proposta comparando à ResNext-101, para diferentes tamanhos de lote (batch size). Entende-se como sendo um lote a quantidade de dados necessários para executar a classificação de gestos para uma câmera, assim um lote de tamanho 2 representa o desempenho como se o *hardware* estivesse processando quadros de duas câmeras concomitantemente. A performance da rede proposta consegue executar de 4.2 a 9.8 vezes mais quadros no mesmo intervalo de tempo para o hardware utilizado, dependendo do tamanho do lote. Além disso, devido à técnica proposta não utilizar todos os quadros disponíveis no conjunto de dados, o hardware tem um maior tempo disponível entre predições para executar a computação necessária (6FPS para o Jester e nvGesture e 15 FPS para o EgoGesture), se comparado a técnicas que utilizam-se da amostra de vídeo capturada na taxa de quadros oferecida pelo conjunto de dados (12FPS para o Jester e 30FPS para o nvGesture e EgoGesture). Considerando que a técnica proposta utiliza apenas uma fração do tempo útil necessário para a processar o quadro quando comparado a tecnica ResNext-101, pode-se alternativamente reduzir a frequência de relógio da unidade de processamento para se otimizar o consumo de energia (ou utilizar-se de *hardware* menos complexo) ao mesmo tempo em que se garante que a janela de tempo dedicada para processar um quadro em tempo real não será violada.

V. Conclusão

Este artigo apresentou uma estrutura de rede neural para reconhecimento de gestos *online*, otimizada para sistemas embarcados, capaz de detecção antecipada e com ativação única para cada gesto. Essas características são essenciais para a aplicabilidade de técnicas de reconhecimento de gestos em aplicações práticas, utilizando sistemas embarcados.

O modelo proposto foi analisado em 3 bancos de dados, estudando a performance da rede quando comparada com outras técnicas *offline* e os resultados que foram obtidos ao adaptar essa rede em ambientes *online*.

O modelo propõe uma característica de *late-fusion* (fusão tardia) dos quadros para que se possa reutilizar um maior volume de dados entre ciclos de predição. Assim, embora a técnica proposta obteve acurácias significativamente menores (dependo do conjunto de dados) que outras técnicas, ao comparar com técnica similar do estado-da-arte para reconhecimeto *online* de gestos, obteve uma tempo de computação médio seis vezes menor.

AGRADECIMENTOS

Este trabalho foi realizado com os apoios da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e NVIDIA.

REFERÊNCIAS

[1] Y. Niu, D. Zou, Y. Niu, Z. He, and H. Tan, "A breakthrough in Speech emotion recognition using Deep Retinal Convolution Neural Networks," pp. 1–7, 2017. [Online]. Available: http://arxiv.org/abs/1707.09917

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, jan 2015. [Online]. Available: https://doi.org/10.1007/s10462-012-9356-9

[3] N. Zengeler, T. Kopinski, and U. Handmann, "Hand gesture recognition in automotive human–machine interaction using depth cameras," *Sensors*, vol. 19, no. 1, p. 59, Dec 2018. [Online]. Available: http://dx.doi.org/10.3390/s19010059

[4] C. R. Salamea Palacios, L. F. D'Haro, and R. Córdoba, "Language recognition using neural phone embeddings and rnnlms," *IEEE Latin America Transactions*, vol. 16, no. 7, pp. 2033–2039, July 2018.

[5] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*. London, UK, UK: Springer-Verlag, 1999, pp. 319–. [Online]. Available: http://dl.acm.org/citation.cfm?id=646469.691875

[6] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," pp. 1–11, 2014. [Online]. Available: http://arxiv.org/abs/1406.2199

[7] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, "Real-time hand gesture detection and classification using convolutional neural networks," *CoRR*, vol. abs/1901.10323, 2019. [Online]. Available: http://arxiv.org/abs/1901.10323

[8] S. Bianco, R. Cadène, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *CoRR*, vol. abs/1810.00736, 2018. [Online]. Available: http://arxiv.org/abs/1810.00736

[9] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek, "VideoLSTM Convolves, Attends and Flows for Action Recognition," *CoRR*, vol. abs/1607.0, 2016. [Online]. Available: http://arxiv.org/abs/1607.01794

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[11] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4207–4215.

[12] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 1–7.

[13] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *CoRR*, vol. abs/1406.2, 2014. [Online]. Available: http://arxiv.org/abs/1406.2199

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] "THE 20BN-JESTER DATASET V1," 2017. [Online]. Available: https://20bn.com/datasets/jester/v1

[16] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "Egogesture: A new dataset and benchmark for egocentric hand gesture recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, May 2018.

[17] T. P. Nguyen, C. C. Pham, S. V. U. Ha, and J. W. Jeon, "Change Detection by Training a Triplet Network for Motion Feature Extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–14, 2018.

[18] O. Köpüklü, N. Köse, and G. Rigoll, "Motion Fused Frames: Data Level Fusion Strategy for Hand Gesture Recognition," 2018. [Online]. Available: http://arxiv.org/abs/1804.07187

Fábio Brandolt Baldissera obtained his Bachelor's Degree in Electrical Engineering from the Federal University of Santa Maria (UFSM), Brazil, in 2016. Currently, he is a Master's Candidate at the Catholic University (PUCRS), Brazil.

Fabian Vargas obtained his Ph.D. Degree in Microelectronics from the Institut National Polytechnique de Grenoble (INPG), France, in 1995. At present, he is Full Professor at the Catholic University (PUCRS), Brazil. His main research domains involve the HW-SW co-design of system-on-chip (SoC) and embedded systems having in mind test, fault-tolerance and reliability considerations for critical applications. Prof. Vargas is researcher of the Brazilian National Science Foundation (CNPq) since 1996. He received the Meritorious Service Award of the IEEE Computer Society for co-founding and chairing the IEEE Latin American Regional TTTC Group and the IEEE Latin American Test Symposium (LATS). Prof. Vargas is Senior Member of IEEE and Golden Core Member of the IEEE Computer Society.