UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

FELIPE FRANCO COSTA

CLASSIFICAÇÃO DE AÇÕES EM VÍDEOS POR MEIO DE REDES NEURAIS CONVOLUCIONAIS BASEADAS EM GRAFOS

DISSERTAÇÃO

CORNÉLIO PROCÓPIO
2020

FELIPE FRANCO COSTA

CLASSIFICAÇÃO DE AÇÕES EM VÍDEOS POR MEIO DE REDES NEURAIS CONVOLUCIONAIS BASEADAS EM GRAFOS

Dissertação apresentado(a) como requisito parcial à obtenção do título de Mestre(a) em Informática, do Programa de Pós-Graduação em Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Pedro Henrique Bugatti

CORNÉLIO PROCÓPIO

2020

Dedico este trabalho a minha família, esposa e
aos meus amigos, pelos momentos de ausência.

AGRADECIMENTOS

Sou grato antes de tudo ao meu Orientador Prof. Dr. Pedro H. Bugatti pela paciência e ensinamentos durante todo percurso. A toda a minha família, a minha mãe Ana Costa e meu pai Cesar Costa, por incentivarem sempre a perseverança, a minha esposa Larissa Bazzo por carinho e incentivo. Sou grato a todos os amigos de pesquisa e professores da UTFPR, em especial a Profª. Drª. Priscila M. T. Saito pelo apoio e colaboração ao conhecimento. Esse trabalho não poderia ser concluído sem todo apoio e carinho de todo os envolvidos.

RESUMO

COSTA, Felipe Franco. **CLASSIFICAÇÃO DE AÇÕES EM VÍDEOS POR MEIO DE REDES NEURAIS CONVOLUCIONAIS BASEADAS EM GRAFOS**. 2020. 58 f. Dissertação (Mestrado em Informática) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Métodos para classificação de vídeos têm evoluido por meio de propostas baseadas em arquiteturas *end-to-end* para aprendizagem profunda. Diversos trabalhos da literatura têm corroborado que tais modelos *end-to-end* são eficazes para o aprendizado de características intrínsecas às imagens (ou *frames* de um vídeo), quando comparados a descritores tradicionais (*handcrafted*). Assim, de maneira geral, utiliza-se redes neurais convolucionais para realizar o aprendizado profundo em vídeos. Quando aplicadas a tais contextos as mesmas podem apresentar variações baseadas em informações temporais, em células de memória (e.g. *long-short term memory* - LSTM) ou até mesmo métodos de entrada de fluxo óptico para auxílio de convolução. Porém, apesar de serem, de certa forma, eficazes para a classificação de vídeos, as mesmas negligenciam a análise global de vídeos, aceitando apenas alguns poucos frames por lote de processamento para treino e inferência. Além disso, não consideram o relacionamento semântico entre diferentes vídeos pertencentes a um mesmo contexto para auxiliar o processo de classificação. Dessa forma, o presentre trabalho visa preencher essas lacunas existentes. Para tanto, serão utilizados conceitos de agrupamento de informação e detecção contextual por meio de redes convolucionais baseadas em grafos (*graph convolutional networks*). Por meio de tal arquitetura espera-se propor um método capaz de criar e explorar o relacionamento entre diferentes vídeos de um dado contexto, visando melhor eficácia quando comparadao aos métodos do estado da arte.

Palavras-chave: Aprendizado Profundo. Redes de Grafos Convolucionais. Visão Computacional. Classificação de Ação.

ABSTRACT

COSTA, Felipe Franco. **VIDEO ACTIONS CLASSIFICATION THROUGH GRAPH-BASED CONVOLUTIONAL NEURAL NETWORKS**. 2020. 58 p. Dissertation (Master's Degree in Computing) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Video classification methods have been evolving through proposals based on end-to-end architectures for deep learning. Many academic works have validated that such end-to-end models are effective for the learning of characteristics intrinsic to videos, especially when compared to traditional, handcrafted, descriptors. In general, convolutional neural networks are used for deep learning in videos. When applied to such contexts, the networks can display variations based on temporal information, based memory cells (e.g. long-short term memory), or even optical flow techniques used in conjunction with the convolution process. However, despite its effectiveness, those methods neglect global analysis, processing only a small quantity of frames in each batch during the learning and inference process. Moreover, they also completely ignore the semantic relationship between different videos that belong to the same context. Thus, the present work aims to fill the existing gaps by using concepts of information grouping and contextual detection through graph-based convolutional neural networks (GCN). With these architectures we hope to propose new approaches to create and explore the relationship between different videos of a given context, improving the state-of-the-art in the process.

Keywords: Deep Learning. Graph Convolutional Network. Computer Vision. Action Classification.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

SIGLAS

CNN	Rede Neural Convolutiva, do inglês *Convolutional Neural Network*
CPU	Unidade Central de Processamento, do inglês *Central Processing Unit*
GCN	Rede de Grafos Convolutivos, do inglês *Graph Convolutional Network*
GPU	Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit*
LRCN	Redes Convolucionais Recorrentes de Longo Prazo, do inglês *Long-Term Recurrent Convolutional Networks*
LSTM	Memória de Longo Prazo, do inglês *Long-Short Term Memory*
MLP	Perceptron Multicamadas, do inglês *Multilayer Perceptron*
RNN	Rede Neural Recorrente, do inglês *Recurrent Neural Network*
UCF	Centro de Pesquisa em Visão Computacional, do inglês *Center for Research Computer Vision*

SUMÁRIO

1 INTRODUÇÃO

Os vídeos digitais são modelos de dados fundamentais que descrevem objetos, símbolos, grafias e até mesmo ações dentro de uma série de imagens que evoluem no tempo. Com a evolução de recursos de *hardware*, foi possível a utilização de algoritmos propostos que eram até então custosos para arquiteturas convencionais de CPUs. Com o advento da computação paralela em arquiteturas GPUs, as mesmas tornaram-se viáveis e interessantes de forma que todo cálculo vetorial presente nos algoritmos envolvendo aprendizado de máquina profundo (*deep learning*) pode ser realizado de forma paralelizada nos núcleos simplificados e numerosos das GPUs (KRIZHEVSKY *et al.*, 2012).

Um dos maiores avanços em *deep learning* para reconhecimento de padrões em imagens foi o uso de filtros convolucionais capazes de aprender a extrair características profundas de forma altamente tolerante a ruídos, distorções e translações nas imagens. A arquitetura de rede chamada Rede Neural Convolucional (CNN) em termos de robustez e de precisão foi um novo nível alcançado no estado da arte, bem como é amplamente utilizada na área de aprendizado de máquina para área de visão computacional (LECUN *et al.*, 1998).

Uma CNN aplica camadas de convolução para extrair características da imagem, gerando mapas que amplificam canais de cores e reduzem as dimensões espaciais da imagem para que uma camada densa, totalmente conectada (ROSENBLATT, 1958) realize a classificação. Essa forma de aprender a extrair características por filtros convolucionais é diferente do que vinha sendo realizado por meio de características específicas a um determinado domínio (*handcrafted*).

Por meio das CNNs, nos últimos anos, o reconhecimento de ações em videos culminou na proposta de várias técnicas. A maior dificuldade de trabalhar com videos é a variável temporal que deve ser levada em consideração. Esse fator leva a diversos problemas como mudanças de posição de câmera, de iluminação, variância de informação por evolução de *frames*, entre outras complexidades que a adição do tempo agrega nas relações dessas imagens estáticas sequenciais.

Atualmente, as características semânticas da mudança na variável temporal de vídeos são trabalhadas de várias formas em aprendizado profundo. Existem soluções que casualmente dependem de redes CNN para extrair características espaciais de imagens, as quais são utilizadas como entrada de modelos que tem características de aprendizado temporal. Outras técnicas aplicam estruturas de memórias de longo prazo (HOCHREITER; SCHMIDHUBER, 1997), ou até mesmo utilizam CNNs distintas para imagens estáticas em conjunto com fluxo ótico (*optical*

flow) (SIMONYAN; ZISSERMAN, 2014).

No entanto, todos os métodos atuais não consideram o contexto de diversos vídeos e o relacionamento entre os mesmos para auxiliar o processo de classificação. Dessa forma, o presente trabalho visou propor um método para justamente realizar tal agregação de informação e alcançar melhorias na classificação de vídeos.

1.1 OBJETIVOS

O objetivo do presente trabalho visa a melhoria da classificação de ações em vídeos utilizando redes convolucionais baseadas em grafos (**??**), agregadas às características extraídas por meio de arquiteturas CNNs. As arquiteturas CNNs foram responsáveis por extrair características profundas das imagens. O intuito almejado foi obter uma melhora significativa no reconhecimento de ações em vídeos, descartando o uso de sistemas complexos baseados em convoluções tridimensionais como as aplicadas em (JI *et al.*, 2013), balanceando a simetria entre eficiência e eficácia.

1.2 ORGANIZAÇÃO DO TEXTO

O presente trabalho está organizado da seguinte forma:

- O Capítulo 2 abrange os principais conceitos necessários para o entendimento do trabalho. São apresentados os principais tipos de arquiteturas de redes neurais convolucionais, as representações de redes convolucionais baseadas em grafos, bem como abordagens significativas de aprendizado profundo relacionados ao presente trabalho e suas idealizações na área de visão computacional;
- O Capítulo 3 aborda a proposta do presente trabalho explicitando o método proposto aplicado para atingir os objetivos elencados. Além disso, são apresentados os experimentos realizados e os resultados obtidos;
- Por fim, o Capítulo 4 aborda as conclusões relacionados ao presente trabalho, bem como explicita possíveis trabalhos futuros visando a melhoria do méotodo proposto. Além disso, também apresenta as publicações oriundas do mesmo.

2 CONCEITOS RELACIONADOS

A área de visão computacional trabalha a inferência com objetos baseados em imagens, tais como *frames* em vídeos. Uma das ramificações da área é a classificação de ações em vídeos, tratada normalmente por dois processos. O primeiro embasa-se na extração de características espaciais ou temporais dos frames representados por imagens, onde são gerados os vetores de características para que, posteriormente, a segunda etapa que trata da classificação consiga ser executada pautado nas características.

O presente trabalho tem como foco modelos profundos que realizam esses dois processos (extração e classificação) de forma concomitante e geram um aprendizado automático. Assim, o presente capítulo será dedicado a abordagens de modelos profundos, capazes de realizar tal processo. As mesmas são utilizadas na área de visão computacional como as redes neurais convolucionais e as baseadas em grafos.

2.1 REDES NEURAIS

O cérebro é um sistema de processamento não-linear e paralelizado, com capacidade de organizar suas estruturas neurais de forma a realizar atividades (e.g. reconhecimento de padrões, percepção e controle motor) muito mais rápido que qualquer computador digital (HAYKIN, 2001). Uma das maiores curiosidades, e ainda em aberto, refere-se ao funcionamento do cérebro humano relacionado a processamentos visuais em imagens estáticas (MARR, 1982), e a função do sistema visual relacionado ao reconhecimento baseado na percepção.

2.1.1 Neurônios Artificiais

O cérebro apresenta de maneira inata a capacidade de estruturar regras por meio de estímulos, como por exemplo estímulos visuais. Tais estímulos são processados e acumulados com o passar do tempo estabelecendo conexões físicas neuronais. O cérebro é de forma geral maleável, ou seja, consegue criar conexões que aprendem padrões para adaptar-se ao meio ambiente por troca de estímulos sensoriais e motores de resposta.

Essas estruturas neurais são formadas a partir de neurônios que são interligados de forma altamente paralelizada e distribuída, regidas por pesos sinápticos que regulam sua intensidade de

excitação de uns para outros. A adaptação desses pesos conectivos modela o aprendizado e a forma que a informação trafega (HAYKIN, 2001). Assim como no cérebro humano, o modelo de neurônio artificial de (MCCULLOCH; PITTS, 1943), denominado de perceptron, tenta emular matematicamente o processo realizado pelo neurônio biológico.

Um neurônio artificial é o coração de uma rede neural artificial, representado como um modelo matemático que aceita entradas e gera uma determinada saída dependente de determinados estímulos (entradas). Na Figura 1 é apresentado um neurônio artificial.

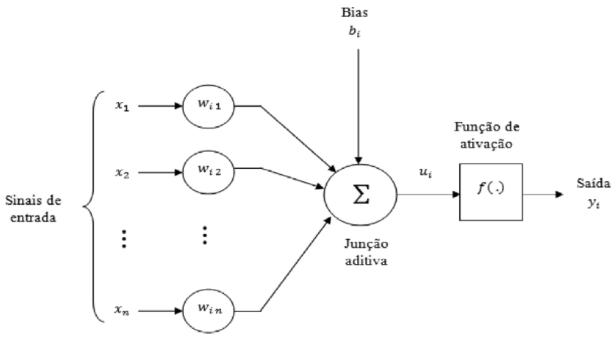

Figura 1 – Exemplo de um neurônio artificial não-linear.
Fonte: (HAYKIN, 2001)

A Equação 1 apresenta a formalização da modelagem matemática utilizada para representar um neurônio artificial. O peso formalizado como ω_{kn} define um fator multiplicativo entre uma entrada x_n, onde o índice n representa a sinapse conectada ao neurônio k. Já b_k representa o viés (*bias*), um fator que aumenta ou diminui a entrada líquida da função de ativação, a qual é representada por $f_{(.)}$ e é uma função qualquer que tem por objetivo adicionar certa não-linearidade às saídas. Na Equação 1 é definido o somatório de todas as entradas utilizadas para gerar a saída por meio do neurônio artificial:

$$u_k = \sum_{j=1}^{m} \omega_{kj} x_j \tag{1}$$

onde m é a quantidade de sinais de entrada; ω_{kn} é o peso que transforma as entradas de forma linear, sendo que tal peso é ajustado em um modelo de treinamento. Uma saída é expressa por u_k que define um modelo com mais de um neurônio, sendo o *k-ésimo* neurônio. Assim como na Equação 1, pode ser tomado o resultado de $u_k + b_k$ como ν_k. Para "quebrar" a estrutura linear, é inserido uma equação não-linear de ativação definida pela Equação 2.

$$y_k = \varphi(u_k + b_k) \tag{2}$$

na qual b_k é o *bias* referente ao neurônio k e φ é a função de ativação. Produzindo uma saída não-linear. Ou substituindo com a notação ν_k, ficando com:

$$y_k = \varphi(\nu_k) \tag{3}$$

2.1.2 Funções de ativação

Essas funções são responsáveis por "quebrar" uma linearidade de cada campo induzido ν resultante de uma operação $Y = XW + B$ das matrizes de pesos, entradas e bias, para que o neurônio propague uma saída não-linear.

Existem diferentes funções matemáticas que podem ser aplicadas para a "quebrar" de linearidade. A primeira função introduzida no *modelo de McCulloch-Pitts* é chamada de *propriedade tudo-ou-nada*, descrita em (MCCULLOCH; PITTS, 1943). Esse tipo de função caracteriza-se por uma função degrau descrita pela Equação 4.

$$\varphi(\nu) = \begin{cases} 1, & \text{se } \nu_k \geq 0 \\ 0, & \text{se } \nu_k < 0 \end{cases} \tag{4}$$

Similar à função degrau utilizada, existe uma variação que possui um intervalo entre 0 e 1, denominada *Função Linear por Partes*, significando que ela é linear em apenas um intervalo, podendo ser descrita no seu todo como uma função não-linear pela sua forma definida na Equação 5.

$$\varphi(\nu) = \begin{cases} 1, & \text{se } \nu_k \geq a \\ \nu, & \text{se } a > \nu_k > b \\ 0, & \text{se } \nu_k \leq b \end{cases} \tag{5}$$

onde a e b são os intervalos que definem em qual período de ν_k os valores são lineares e em um intervalo de (a,b). Uma das funções mais utilizadas para neurônios perceptron é a *Função Logística*, definida como uma *Função Sigmoidal* não linear (ver Equação 6).

$$\varphi(\nu) = \frac{1}{1 + e^{-a\nu}} \tag{6}$$

Embora as funções citadas tenham tido seu momento de grande uso em redes neurais, atualmente uma das mais utilizadas contando com suas variações, é a denominada *Unidade*

Linear Retificada (ReLU). A mesma é definida formalmente na Equação 7.

$$\varphi(\nu) = \max(0,\nu_k) \tag{7}$$

Segundo a Equação 7 a saída será 0 para valores de ν_k negativos ou ν_k para maiores que zero, logo, é possível observar que para essa função há uma vantagem de não ativar todo os neurônios simultaneamente. Isso ocorre pois para valores menores que zero ela propaga 0 para neurônios consecutivos. Com a ressalva de que essa função é utilizada apenas em camadas ocultas.

2.1.3 Aprendizagem por Correção de Erro

O ajuste dos pesos em um neurônio é feito de forma a acrescentar uma taxa ($\Delta\omega$), a qual é definida formalmente pela Equação 8 adicionando a ω um valor que faz com que a distância entre a saída desejada e a saída inferida seja a menor possível.

$$\omega(t+1) = \omega_t + \Delta\omega \tag{8}$$

Os algoritmos de aprendizado contam com otimizadores para incremento do vetor de pesos, no tempo discreto de execução do aprendizado dado por n, o qual compara uma saída desejada $d_k(n)$ com uma saída predita $y_k(n)$, definida na Equação 9.

$$\epsilon = d_k(n) - y_k(n) \tag{9}$$

A função de custo associada ao peso do neurônio k é representada por $\xi(\omega)$. Esse custo (i.e. erro) é utilizado para atualizar um peso juntamente a uma taxa de aprendizado definida como η com característica fixa chamada de *Regra de Adaptação com Incremento Fixo*. O processo de otimização produz uma correção nos pesos ω com a minimização da função custo $\xi(\omega)$.

Existem na literatura diferentes algoritmos de otimização passíveis de serem aplicados a tal contexto, bem como funções de custo (*loss functions*). Um exemplo de uma função de custo trivial aplicada em diversos trabalhos é baseada na minimização do erro quadrático médio (ver Equação 10).

$$\xi(n) = \frac{1}{2}e_k^2(n) \tag{10}$$

Essa função custo calcula o valor da energia do erro quadrático para maximizar a penalidade para grandes distâncias/divergências entre o resultado esperado e o predito. O ajuste de ω é dado pela Equação 11, sendo que o peso é atualizado somando o anterior ao novo *Delta* por meio da Equação 12.

$$\Delta\omega_{kn}(n) = \eta e_k(n) x_j(n) \tag{11}$$

$$\omega_{kn}(n+1) = \omega_{kn}(n) + \Delta\omega_{kn}(n) \tag{12}$$

2.2 APRENDIZADO PROFUNDO

Aprendizado profundo é uma área de aprendizado de máquina. O mesmo apresenta uma forma hierárquica de modelar abstrações profundas que são transmitidas de camada a camada por meio de transformações lineares ou não-lineares (GOODFELLOW *et al.*, 2016; DENG; YU, 2014).

Essas operações em conjunto com a otimização de uma função objetivo de classificação são o que tornam viáveis a abstração de características nas várias camadas densas de um modelo. Dessa forma, trata-se de uma abordagem que aprende diferentes representações hierárquicas que são repassadas adiante camada por camada (DENG; YU, 2014).

2.2.1 Redes Neurais Convolucionais

Uma rede neural convolucional (*convolutional neural network*) é uma rede perceptron de múltiplas camadas projetada especificamente para reconhecer formas bidimensionais com alto grau de invariância quanto a translação, deslocamento, inclinação e outras formas de distorções. Esta difícil tarefa é aprendida de forma supervisionada por meio de uma rede cuja estrutura é definida em filtros (*kernels*) de convolução no qual seu papel é extrair características locais e passar adiante essa informação.

As saídas do processo de convolução por meio de tais filtros são denominadas de *mapas de características*. Cada camada de uma CNN é composta por vários desses mapas. Os mapas possuem pesos sinápticos independentes e restritos a compartilhar o mesmo conjunto de pesos, que permite que esse tipo de arquitetura reduza a quantidade de parâmetros livres. Os mapas de características são vetores que carregam informações espaciais das imagens estáticas,

convencionalmente extraídos por redes CNN, já que esses são eficientes em extrair características espaciais e de cores como ilustrado na Figura 2.

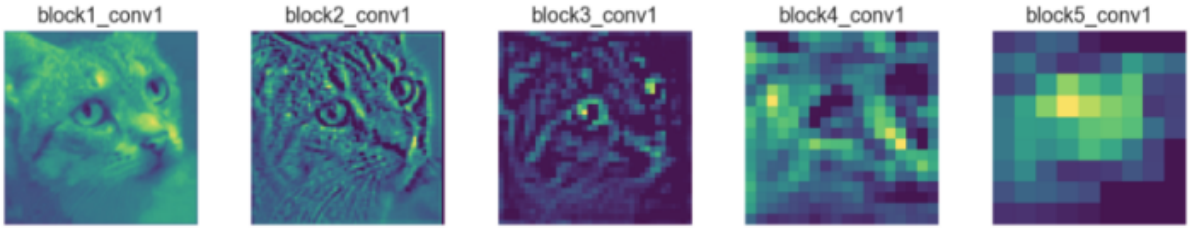

Figura 2 – Exemplo de como filtros de convoluções agem nas imagens desde a entrada até o mapa de característica gerado na última camada de convolução

A rede também conta com camadas de *subamostragem*, que extraem de cada mapa de característica a metade do seu tamanho espacial por algum critério de exclusão de características. É fácil perceber que uma rede neural convolucional conta com mais de uma técnica na sua arquitetura, como é visto na Figura 3, a qual ilustra a arquitetura CNN denominada LeNet-5 (LECUN *et al.*, 1998).

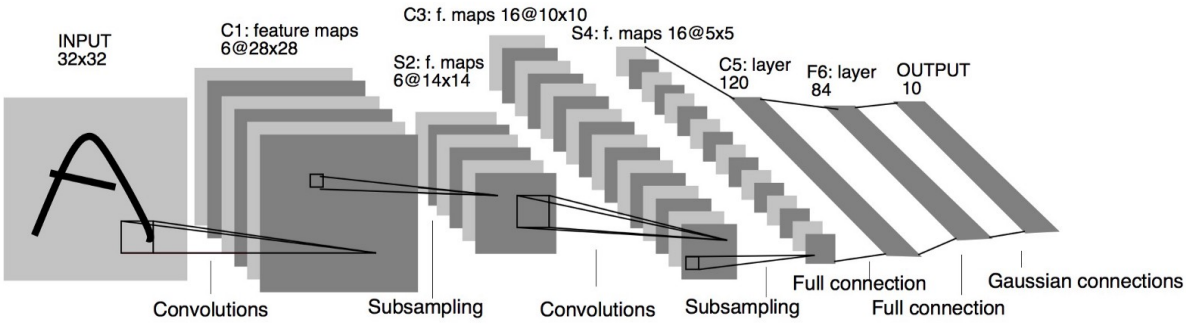

Figura 3 – Arquitetura da LeNet-5, uma rede neural convolucional para reconhecimento de dígitos escritos à mão.

Fonte: (LECUN *et al.*, 1998)

2.2.1.1 Convolução

Em análise funcional de sinais, uma convolução $h(x)$ é o resultado composto por duas funções que resultam em uma terceira, medindo a soma do ponto submetido a superposição de área das duas funções em um deslocamento definido de tempo $(t - \tau)$, e considerando duas funções, uma por $f(\tau)$ e outra definindo um sinal que desloca ao longo do tempo $g(t - \tau)$ e formalizadas em cálculo infinitesimal por meio da Equação 13.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau) d\tau \tag{13}$$

definido no exemplo da Figura 4.

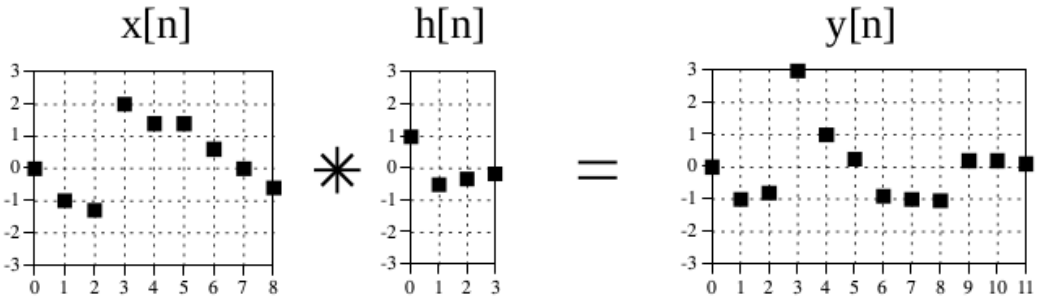

Figura 4 – Um sinal de entrada de nove pontos, convoluido com uma resposta ao impulso de quatro pontos, resultando em um sinal de saída de doze pontos.

Fonte: (SMITH, 1997)

Em situações discretas, que de fato é a intenção do presente trabalho, a convolução é definida na Equação 14 como *produto de Cauchy* $h(k)$:

$$h(k) = \sum_{j=0}^{k} f_j \cdot g_{(k-j)} \tag{14}$$

Uma vez definido formalmente o que é uma convolução, é razoável explicar que em redes convolucionais o filtro de convolução é definido como um peso de três dimensões (c, s_1, s_2), onde c é a quantidade de canais necessários. Cada canal c possui duas dimensões espaciais de s_1 e s_2 de forma que elas já se explicam.

Quando é inserida uma entrada $X_{(i,j)}$ (*e.g.*, uma imagem com duas dimensões espaciais (i,j) para cada canal de cor c) os pesos têm o mesmo formato na dimensão c que os canais de cores da imagem, e dimensão (s_1,s_2) definindo o tamanho de filtro que será deslizado sobre $X_{(i,j)}$ e gerando um resultado convoluído dado pela Equação 15.

$$u^k_{(i,j)} = \sum_{t=0}^{c} \sum_{p=0}^{s_1} \sum_{q=0}^{s_2} W^{k,t}_{(p,q)} \times X^t_{(i+p,j+q)} \tag{15}$$

Produzindo uma convolução na região (i,j) de X, aplicando então uma função de ativação $\varphi(\cdot)$ em u (Equação 16).

$$f^k_{(i,j)} = \varphi(u^k_{(i,j)}) \tag{16}$$

Esse cálculo é repetido percorrendo a área chamada de mapa de características (*e.g.*, uma imagem de entrada com dimensão (i,j)), de forma a gerar um novo mapa de características. Então, a convolução é feita para toda entrada X como ilustrado Figura 5.

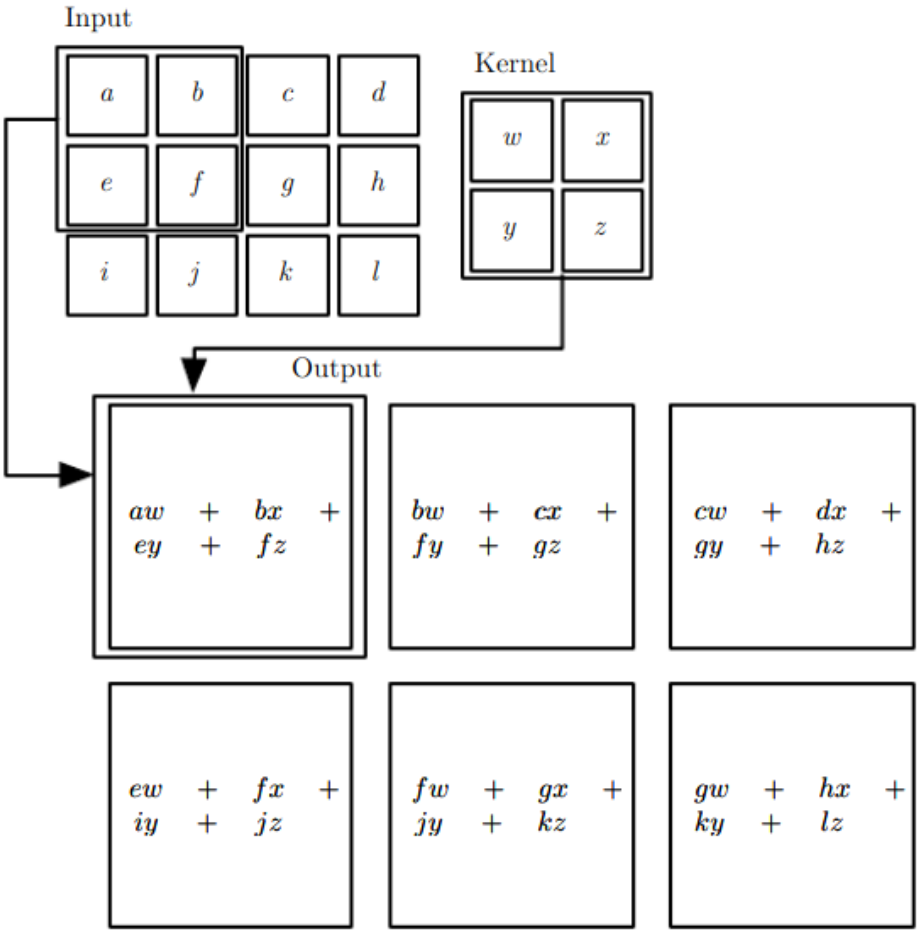

Figura 5 – Um exemplo de convolução 2-D com a saída restringida apenas a posições onde o kernel está inteiramente dentro da imagem, chamado de convolução "válida" em alguns contextos.

Fonte: (GOODFELLOW *et al.*, 2016)

Nota-se que no exemplo da Figura 5 é definida uma convolução sem restrição de deslizamento do filtro sobre a entrada de letras, bem como nem mesmo é definida uma solução para que a borda do tensor de entrada não seja violada. Em técnicas de redes convolucionais, esses tipos de parâmetros são os de *stride* e *valid padding*.

O *stride* define quantas posições por vez o filtro se move nos índices de $X_{(i,j)}$, podendo este reduzir o tamanho dos mapas de características (*i.e.*, profundidade de cores) gerados pela convolução. Já o *valid padding* é o tamanho do tensor X como limite de ação do filtro convolutivo, se respeitado, o filtro nunca desliza para fora do tensor. Caso seja necessário capturar convoluções na margem precisa-se acrescentar valores nulos e aumentar as dimensões do tensor de entrada.

2.2.1.2 Pooling

Na operação de *pooling*, o objetivo é remover a sensibilidade da rede a pequenos ruídos, visto que a operação diminui as características de uma determinada região em conjunto menor, bem como diminui a redundância da rede (FUKUSHIMA, 1980).

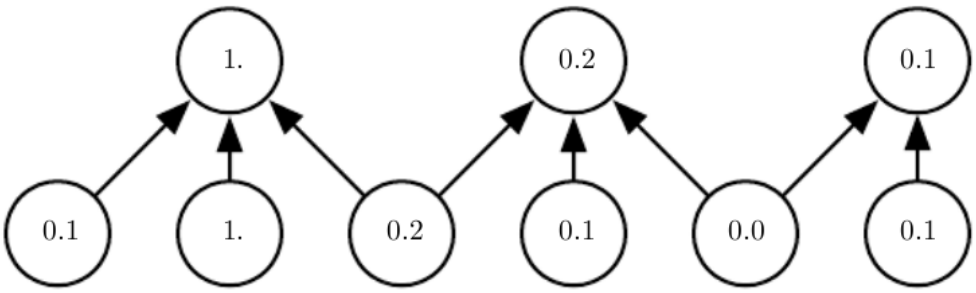

Figura 6 – Exemplo de uma forma de pooling que seleciona o valor mais alto de características e reduzem a dimensão no mapa de características.
Fonte: (GOODFELLOW *et al.*, 2016)

A Figura 6 define a forma mais comum e mais "barata" de *pooling*, onde é considerada, dentre algumas características em uma determinada janela no mapa de características, aquela de valor mais alto. Esse técnica é formalizada na Equação 17.

$$z_i = max\{x_j\} \tag{17}$$

2.2.2 Redes convolucionais baseadas em Grafos

Redes CNN têm sido extremamente bem sucedidas em problemas de aprendizado de máquina em que as coordenadas da representação de dados subjacentes têm uma estrutura tensorial de 1, 2 ou 3 dimensões, bem como os dados a serem estudados nessas coordenadas têm equi-variância translacional e invariância com relação a essa estrutura (BRUNA *et al.*, 2014).

Há duas implicações na equi-variância, a primeira é a equi-variância de medição e a segunda a invariância informal. Por exemplo, uma imagem I qualquer pode ser transladada para I' com coordenadas originais (x_m, y_m) para as novas coordenadas de (x'_m, y'_m), produzido por $(x_m - u$,$y_m - v)$, portanto invariante. Já $m' = m$ é equivariante quanto às medições finais de convolução, por meio de coeficientes geométricos que são os filtros de convolução.

Nas diversas formas em que redes de grafos são construídas, estas são categorizadas

em aprendizado profundo como modelos geométricos. O ideial desses modelos é trabalhar com métricas multidimensionais na forma de tratar a equi-variância proposta por modelos convolutivos. Assim como um filtro de convolução consiste em trabalhar um coeficiente fixo de um aspecto geométrico na matriz, a forma que a Redes de Grafos Convolucionais (*Graph Convolutional Networks* - GCN) trabalha é tentando generalizar esses filtros por normatizações de distância entre pontos (i.e., que são os valores de uma característica qualquer no espaço não-euclidiano representando uma geometria extrínseca).

A arquitetura presente na categoria de modelos geométricos de aprendizado profundo utiliza do potencial de aprendizado automático baseado em filtros de convolução para atacar problemas com abordagem de grafos arbitrariamente estruturados.

Além disso, existe uma dificuldade de generalizar modelos como CNN ou RNN. Alguns trabalhos recentes mostram tais adaptações (DUVENAUD *et al.*, 2015; LI *et al.*, 2016; JAIN *et al.*, 2015), outros trabalham com uma técnica baseada em teoria de grafos espectrais (BRUNA *et al.*, 2014; HENAFF *et al.*, 2015), para definir filtros baseados em CNN clássicas.

O objetivo de redes baseadas em grafos é de aprender um conjunto de sinais de entrada de um grafo $G = (\nu, \varepsilon)$ com uma entrada de matriz X de formato $N \times D$, onde N é o número de nós e D é o número de características e uma matriz A de adjacência. E cada camada produz uma saída Z com formato de $N \times F$ e F representa o número de características por nó. Podendo então ser definido pela Equação linear 18.

$$H^{(l+1)} = f(H^{(l)}, A) \tag{18}$$

com $H^{(0)} = X$ e $Z = H^{(l)}$, de forma que a definição é linear. Uma formalização de $f(H^{(l)}, A)$ é dada a seguir (Equação 19):

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \tag{19}$$

definindo uma aplicação não-linear σ e um peso W por camada l.

Há um problema em multiplicar com uma matriz de adjacência A em que para cada nó é somado os vetores de características de todos os nós vizinhos menos o do próprio nó. Esse problema é contornado pela somatória de uma matriz de identidade I à matriz de adjacência resultando em $\hat{A}$. Outro problema é que a multiplicação por A mudará a escala dos vetores de características, logo, se todas as linhas somam um multiplicando $\hat{A}$ por D^{-1}, que é a matriz de

grau diagonal. Formalizando então a propagação $f(H^{(l)}, A)$ de forma normalizada (veja Equação 20).

$$f(H^{(l)}, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{20}$$

2.3 APRENDIZADO PROFUNDO EM VÍDEOS

A tarefa de aprendizado profundo em vídeos é computacionalmente onerosa e existem alguns problemas relacionados por conta da estrutura adicional de tempo. Apesar de modelos CNN estarem bem estabelecidos quanto a eficiência de extração de características em imagens estáticas, bem como estarem em boa parte da literatura em visão computacional com vídeos, ainda existem os problemas de raciocínio temporal.

Um dos problemas candidatos é o de baixa capacidade dessas redes com sequências grandes de *frames*, nos quais precisam ser inteiramente carregados, e se ausentes podem culminar em falta de informações eficientes para o processo. Em relação a modelos de classificação em vídeo existem duas maneiras de serem feitas, de forma manual quanto à extração das características, ou de forma interna ao modelo profundo.

As técnicas manuais usadas consistem em criar descritores para que classificadores obtenham respostas das possíveis classes. Esses trabalhos se baseiam em determinar novas formas de representatividades com extração de características em vídeos com dimensões compactas, como os contornos de Harris 3D em (LAPTEV, 2005). Já em (WILLEMS *et al.*, 2008) usa-se uma medida Hessiana para saliência que tem proximidade com processos como modelagem de cubóides de (DOLLAR *et al.*, 2005). Outras técnicas como (WANG *et al.*, 2011) usam fluxo óptico para extrair trajetórias densas. Essas características capturadas são passadas normalmente a um histograma como em (LAPTEV *et al.*, 2008).

Em aprendizado profundo para vídeos, diferente de técnicas que são usadas de forma manual, divididas nas etapas de extração de características manualmente e classificação, as arquiteturas *end-to-end* são construídas com objetivo de aprenderem tanto a classificar quanto a descrever os dados de entrada. São várias as técnicas que podem ser utilizadas para trabalhar com vídeos para criar semântica temporal em frames subsequentes de forma interna nesses modelos profundos.

A principal dificuldade de alguns modelos profundos é a já citada restrição em quantidade de frames a serem analisados, a exemplo de redes com convolução 3D (BACCOUCHE

et al., 2011; JI *et al.*, 2013; KARPATHY *et al.*, 2014) que empregam clipes com quadros de imagens para aprender implicitamente padrões em um único tensor representando um vídeo. Em (KARPATHY *et al.*, 2014), os autores demonstram que seu modelo é apenas uma fração melhor do que redes CNN treinando imagens de vídeos de forma individual. A Figura 7 ilustra para tal modelo que as convoluções são empregadas de forma local e limitadas a poucos quadros, por restrições dos enormes tensores a serem alocados na memória de uma única vez.

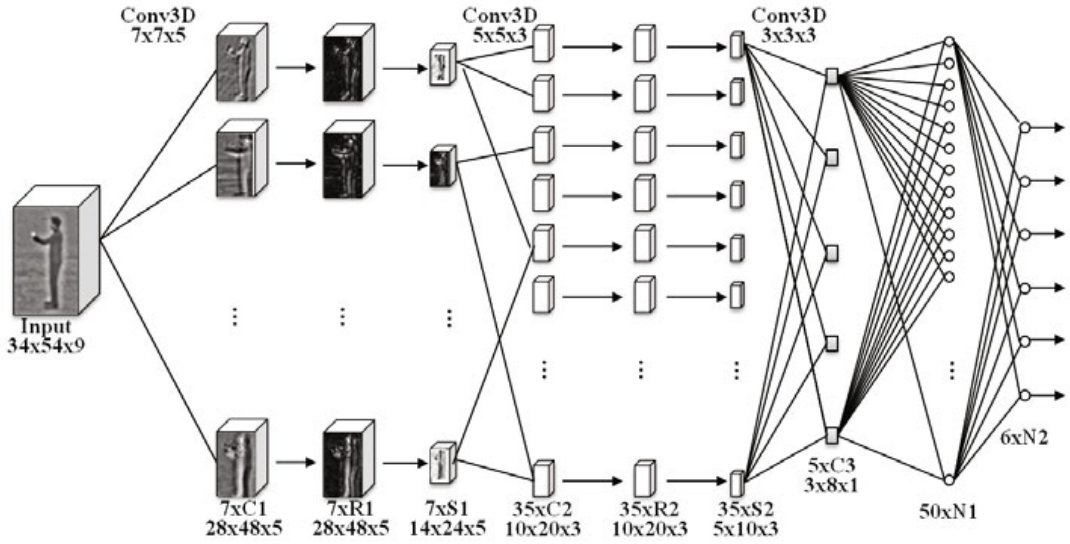

Figura 7 – Extração de caracteristicas de múltiplas imagens em uma convolução dita 3D.
Fonte: (BACCOUCHE *et al.*, 2011)

Algumas outras abordagens podem ser tomadas para transmitir informações temporais para as redes como em (SIMONYAN; ZISSERMAN, 2014), que passa o fluxo óptico como parâmetro de entrada para inferência, limitada a 10 quadros apenas, demonstrado na Figura 8. Esse modelo sofre do mesmo problema de restrições a informações locais de vídeo, onde qualquer informação importante de algum quadro pode se perder, e novamente, não é tão superior a uma abordagem menos complexa de imagens únicas com CNN 2D convencional.

Ao invés de tentar aprender os recursos espaço-temporais limitados a curtos períodos, é consistente o uso de arquitetura CNN para extração das características individuais que são passadas para unidades de memória de longo prazo (LSTM), permitindo entender os padrões estabelecidos conforme o tempo no seu treinamento, fator considerado em (BACCOUCHE *et al.*, 2010). A Figura 9 ilustra tal proposta.

Existem soluções para o não uso de técnicas baseadas em retroalimentação, como empregar soluções arquiteturais ao final da estrutura de *pooling* das redes convolucionais (YUE-HEI NG *et al.*, ; LAPTEV *et al.*, 2008; WANG *et al.*, ; JAIN *et al.*, 2013), onde as características

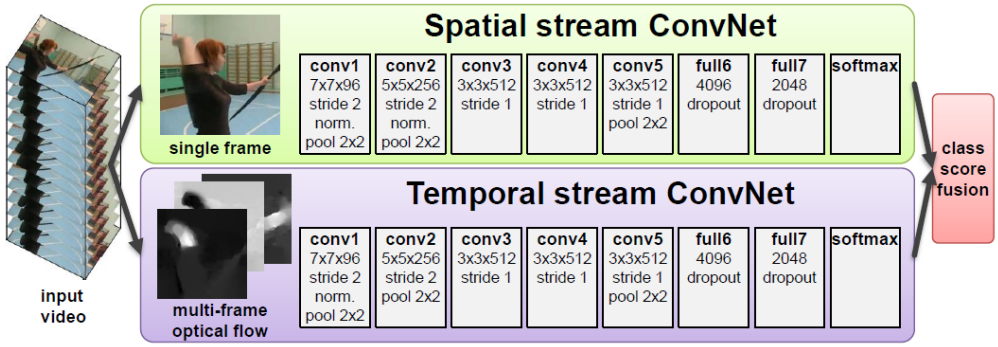

Figura 8 – Exemplo de inferência pelo modelo com adição do fluxo óptico (Two-Stream Net).
Fonte: (SIMONYAN; ZISSERMAN, 2014)

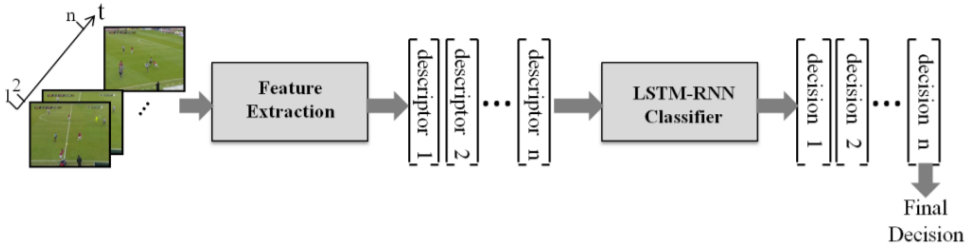

Figura 9 – Unidades LSTM subsequentes de extração de características por CNN.
Fonte: (BACCOUCHE *et al.*, 2010)

de cada imagem são extraídas de forma individual e então a classificação é feita por meio de algumas formas de *pooling* a nível de vídeo, demonstrando um tipo acurácia similar ao modelo com LSTM.

3 MÉTODO PROPOSTO

O presente trabalho teve por objetivo agregar além de descrever visualmente um vídeo (ou *framens* do mesmo) agregar a essas descrições informação de relacionamento entre diferentes *frames* de um determinado vídeo pertencentes a um dado contexto, visando a melhoria da eficácia na classificação de ações.Para tanto, foram utilizadas as arquiteturas convolucionais baseadas em grafos que são basicamente convoluções pela condição de entrada de uma matriz de adjacência, sendo que a convolução é o processo de ativar ou não sinais específicos. Essa matriz de adjacência é normalizada e inserida como um filtro dentro da arquitetura. Como em vídeos essa relação dos nós que devem conectar-se não é explícita, pode aplicar a tais ligações diferentes políticas. Esse fato, também abriu um considerável espectro de possibilidades do presente trabalho. Todos os experimentos foram executados nos limites da GPU, no qual o modelo é um NVIDIA Geforce GTX 1050TI.

3.1 METODOLOGIA

Para alcançar os objetivos do presente trabalho foi desenvolvida uma metodologia a qual é ilustrada na Figura 10. As subseções subsequentes descrevem detalhadamente as etapas da metodologia proposta. Pode-se verificar que a mesma foi estruturada em 5 etapas principais, que são:

1. Aquisição de vídeos e seleção dos *frames* dos vídeos;
2. Extração de características via CNN;
3. Novo banco de dados com características profundas;
4. Construção da matriz de adjacência para representação do grafo;
5. Classificação utilizando GCN.

3.1.1 Aquisição de vídeos

A etapa de aquisição de vídeos abrangeu um *dataset* do estado da arte relacionado ao problema de classificação de ações em vídeos. Ao definir um *dataset* existem diferentes

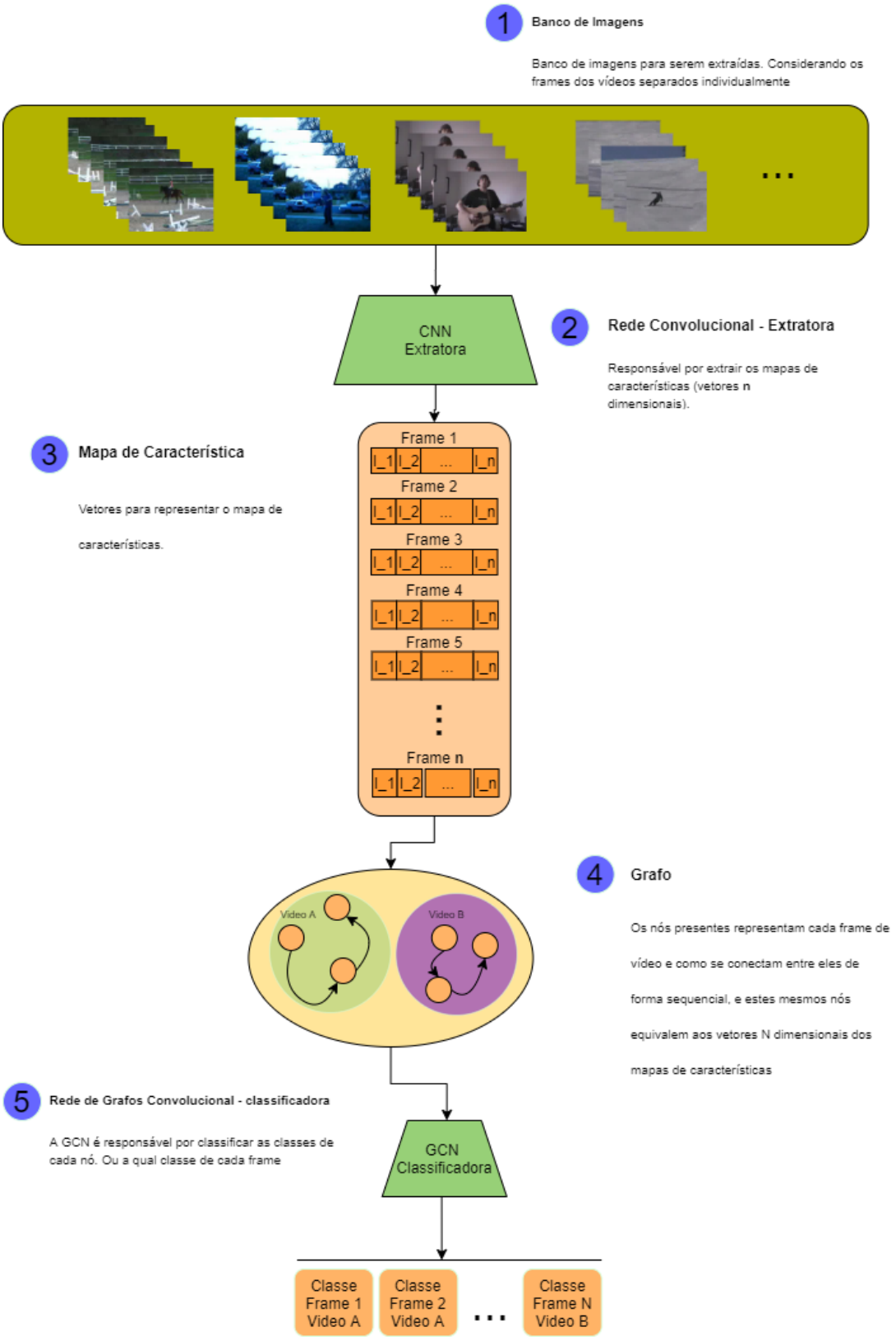

Figura 10 – Pipeline da metodologia proposta

considerações a serem realizadas com relação ao contexto dos mesmos. Para o presente trabalho foi considerado um volume de amostras considerável total e por classe, sendo as mesmas

com pequena variação de cardinalidade. Isso foi estipulado, pois o foco principal era a correta classificação de ações e não a avaliação de robustez do método proposto relacionada a conjuntos de dados com altíssimo desbalanceamento, fato que desviaria o objetivo do trabalho.

Outra consideração a ser feita em relação à escolha do processo de seleção dos vídeos é que optou-se por utilizar vídeos disponibilizados publicamente e obtidos da web. Assim, esses foram gravados em ambientes irrestritos e não controlados, geralmente, com variações de movimento da câmera, várias condições de iluminação, oclusão parcial e quadros de baixa qualidade. O contexto abordado no método proposto foi de acões envovendo entes humanos e objetos. No entanto, isso não impossibilita a aplicação do método proposto em outros contextos por meio de calibrações contextuais intrínsicas.

Além do contexto da aquisição de vídeos entra em questão diferentes políticas para a seleção de *frames* dos referidos vídeos. A política padrão estipulada pelo método proposto, considerando menor custo computacional foi a seleção intervalar. Assim, o intervalo de seleção de *frames* (*hops*) de um dado vídeo foi definido como a razão do total de *frames* no vídeo (n) pela quantidade desejada de *frames* (f).

Dado um conjunto de vídeos $C = v_1, v_2, ..., v_t)$, sendo v_i um dado vídeo composto por n *frames* (i.e., $v_i = r_1, r_2, ..., r_n$), define-se uma quatidade f de *frames* a serem selecionados, sendo que $f \leq n$ (se $f = n$ todos os *frames* são considerados, no entanto o custo computacional é proibitivo e tal possibilidade foi descartada). Dessa forma, a partir do conjunto de vídeos é gerado um novo banco de *frames* (imagens) dos respectivos vídeos, o qual possui cardinalidade $|C| * f$. A Figura 11 ilustra a escolha de determinados *frames* retirados de um *dataset* público da literatura pela política intervalar. Obviamente, existe uma grande quantidade de políticas que podem ser analisadas para a escolha dos *frames*, porém esse não foi o foco central do presente trabalho.

3.1.2 Extração de Características

Para a descrição dos vídeos o método proposto utiliza características profundas (*deep features*) obtidas por meio de arquiteturas CNN, com com aplicação do conceito *transfer learning*, que assimila pesos treinados da arquitetura definida, no qual vem da etapa de treinamento baseado em um conjunto de dados que compartilha das características de um conjunto desejado. O conjunto de dados utilizado nos pesos pré-treinados oriundos do *dataset ImageNet* (RUSSAKOVSKY *et al.*, 2015). Para tanto, basicamente, foi desassociada a camada totalmente conectada

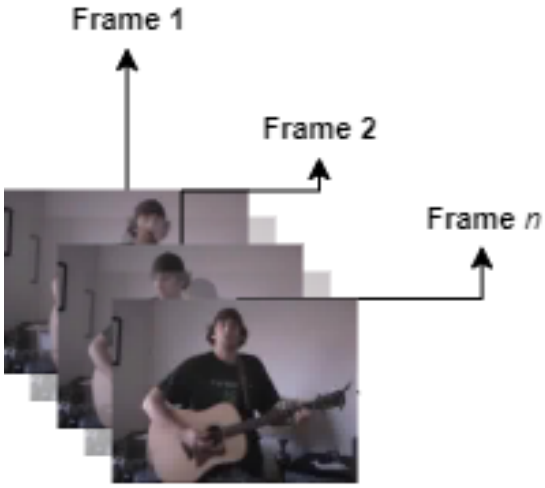

Figura 11 – Seleção de *frames* de cada vídeo para redução e formação do subset de dados. Fonte: (SOOMRO *et al.*, 2012)

(*fully connected*), sendo nada mais que uma camada de classificação MLP. A camada de saídas foi definida como um *averaging pooling* seguida por uma operação de *flatten* para especifiar a mesma em formato de vetor unidimensional com a representação das características. Vale ressaltar que qualquer arquitetura CNN pode ser utilizada no método proposto.

Um exemplo do conceito de *transfer learning* pode ser visto na Figura 12, a qual ilustra a arquitetura CNN denominada ResNet. Na mesma, existe a camada fc com 1000 neurônios, entretanto são os neurônios com o mesmo numero de classes existentes do *dataset ImageNet*, no qual não está presente no modelo usado como extrator. Assim, para obter as características extraídas, basta obter o mapa de caracteristicas unidimensional, por meio da saída da camada de *averaging pooling* e obtendo um vetor n-dimensional.

3.1.3 Construção e Filtros do Grafo

Em problemas de recuperação e agrupamentos a similaridade é fundamental (e.g. definição de um cluster, etc), uma medida da similaridade entre dois padrões é essencial para a maioria desses procedimentos (e.g. agrupamento). Tal medida de distância deve ser escolhida com cuidado (JAIN *et al.*, 1999). No presente trabalho, utilizou-se filtros de grafos estruturados de forma a conectar sequências temporais de imagens que representam um vídeo. Para tanto, foi definida uma sequência cronológica dos *frames*, passando para a rede uma noção temporal definida pela ligação desses *frames* como nós.

Assim como em (**??**), que considera a matriz de adjacência de conexões normalizadas

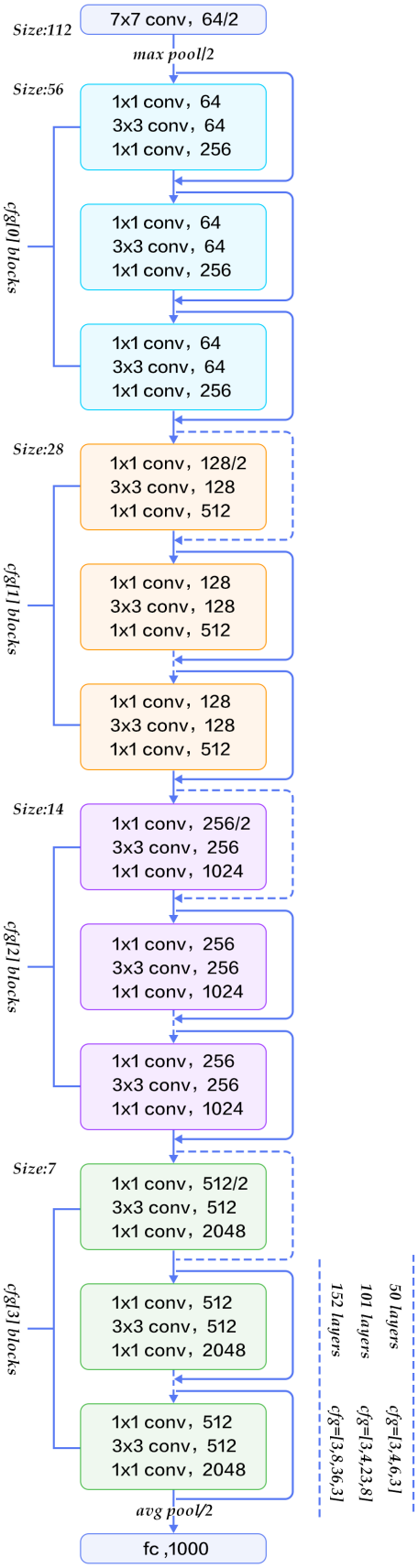

Figura 12 – Ilustração da arquitetura ResNet

como formalmente definido na Equação 20 como próprio filtro de convolução, o presente método segue a mesma estratégia. A Figura 13 ilustra um exemplo de uma GCN onde cada X_i representa um *frame* conectado a outro no contexto da análise de vídeos. No entanto, vale ressaltar que para o método proposto foi considerado que um dado *frame* conecta-se apenas com ele mesmo e com um outro *frame* sempre à frente na dimensão tempo (cronologicamente).

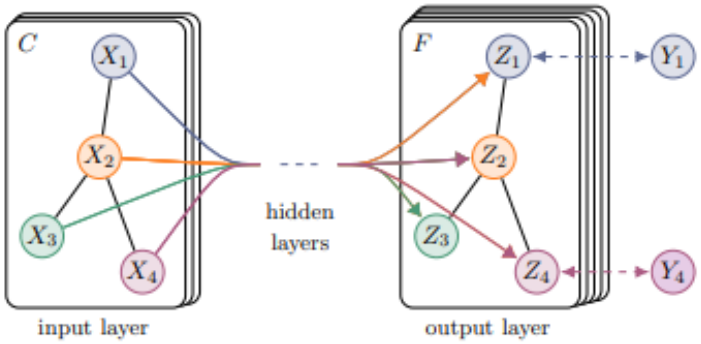

Figura 13 – Arquitetura da GCN para cada X_i representando um *frame* conectado a outro X no contexto de vídeo.

As conexões foram feitas de duas formas. A primeira, não tão fiel às comparações, com outros modelos, pois é uma forma de classificação de *frames* separados, como amostras singulares (um vetor de características para cada frame, sendo que cada um torna-se um nó do grafo) e não o vídeo como uma amostra única (nó do grafo). A segunda forma de considerar as amostras foi realizada por vídeos completos, o total de *frames* presentes no vídeo é o equivalente a uma única amostra (vetores de características dos *frames* selecionados são concatenados representando um nó do grafo). Nas conexões intervalares, cada nó representa um *frame* de vídeo e esses *frames* estão sequencialmente conectados. A exemplo disso é que para cada escolha de f frames, esses são conectados em sequência até que todos estejam conectados, cada um como uma amostra singular, como na Figura 14.

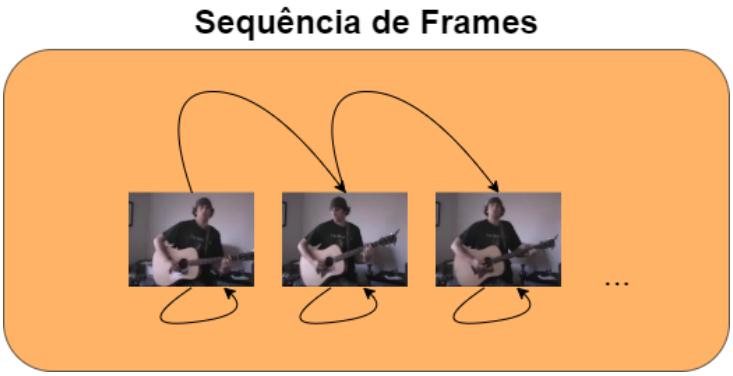

Figura 14 – Sequência de *frames* de cada vídeo conectados.

Na segunda solução, é considerado um vetor n-dimensional que consiste de f *frames* concatenados. Para a conexão entre diferentes vídeos (nós do grafo) foi calculada a distância Euclidiana (Equação 21) entre os vetores de características dos mesmos (q_i e p_i). Posteriomente, para realizar a poda de tais conexões foi considerado um limiar, sendo que arestas com pesos maiores que tal limiar foram podadas.

No presente trabalho, tais limiares foram escolhidos de forma empírica, visando que conexões entre vídeos similares fossem mantidas e caso contrário fossem desligadas. Para tanto, a aresta $e_{(i,j)} = 1$ é definida por $d < limiar$, caso contrário $e_{(i,j)} = 0$, considerando a matriz de adjacências do grafo gerado. Apesar de ter sido definido empiricamente no presente trabalho, o limiar pode ser definido por meio de outras políticas. Por exemplo, seria possível pensar em extrair métricas relacionadas ao grafo e pesos das arestas para definir nós *hub*, entre outros. Porém, esse não foi o foco do presente trabalho, visto que o escopo poderia estender demasiadamente o mesmo.

$$d = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{21}$$

3.1.4 Classificação e Validação

A etapa de classificação utiliza o grafo repassado à GCN para tarefa de inferência de ações por classe de cada vídeo representado por um conjunto de vetores de caracteríticas X. Para tanto, cria-se um tensor de entrada onde serão L nós estruturados no grafo, cada um representado por seu respectivo vetor de características composto por d dimensões. O problema geral é a correlação entre os vetores de características a serem atribuídos nas ligações dos nós. Como em (**??**), esses filtros de grafos são implícitos à rede, os filtros foram adaptados no presente trabalho a fim de definir uma função F_A, sendo A uma matriz de adjacências, que criasse a relação de nós entre as características relacionadas. Como um primeiro passo foi considerado um grafo completo (i.e., totalmente conectado). Posteriormente, as arestas do mesmo foram podadas com intuito de melhor definição das interelações entre os nós quando os mesmos representam vídeos (i.e., concatenação dos vetores de características diferentes *frames* do vídeo).

Para a validacão de todo o processo de classificacão foram realizados testes para demonstrar resultados obtidos. Para tanto, foram utilizadas métricas da literatura como por exemplo: acurácia, matrizes de confusão, precisão (*precision*), revocação (*recall*) e f-score. Além

Tabela 1 – Matriz de Confusão

		Resultado Predito		
		p	**n**	**total**
Valor real	**p′**	Verdadeiro Positivo	Falso Negativo	P′
	n′	Falso Positivo	Verdadeiro Negativo	N′
	total	P	N	

disso, obviamente, o método proposto foi comparado com outros da literatura.

Apenas para ilustrar sucitamente as métricas utilizadas, a matriz de confusão considera as linhas como a inferência das classes das amostras em teste e as colunas estipulam a qual classe as amostras realmente pertencem. A Tabela 1 ilustra uma matriz de confusão para um problema binário (i.e., duas classes).

Por meio da matriz de confusão, pode-se extrair métricas de validação baseado em totais de rotulação (predição) gerada pela rede em comparação com a rotulação real. Considerando a disposição da matriz ilustrada, tem-se então que, a quantidade de elementos quando a linha e a coluna são do mesmo indice é considerado um *verdadeiro positivo* (VP), ao passo que um elemento qualquer disposto em uma linha e coluna com indíces diferentes pode representar um *falso positivo* (FP) ou um *falso negativo*(FN). O *falso negativo* ocorre quando uma amostra de classe real qualquer s é predita pelo modelo como outra. O *falso positivo* ocorre quando é predita uma amostra como de uma classe u, porém a mesma pertence a outra classe real. Por fim, o *(verdadeiro negativo)* (VN) é a somatória das diagonais *verdadeiros positivos* de todas as outras classes com exceção da classe analisada.

A acurácia representa a quantidade de acertos dentro de todas as possibilidades das predições possíveis, demonstrada na Equação 22.

$$\textit{Acurácia} = \frac{TP + TN}{TP + TN + VP + FN} \tag{22}$$

No caso, o método proposto foi comparado e confrontado com outros também por meio das métricas *precision*, *recall*, *f-score*, bem como acurácias por classe e global. A precisão, definida formalmente na Equação 23, é o resultado da proporção de *verdadeiros positivos* de uma determinada classe e o quão bom é esse resultado em relação ao montante de positivos

classificados, obviamente considerando o conjunto de teste.

$$\textit{Precisão} = \frac{VP}{VP + FP} \tag{23}$$

$$\textit{Revocação} = \frac{VP}{VP + FN} \tag{24}$$

Na métrica de *recall*, definida na Equação 24, a mesma estipula o quão bom é um verdadeiro positivo de uma classe pelo total das suas amostras. A métrica *f-score*, descrita na Equação 25, trata-se de uma média harmônica entre *recall* e *precision*. Assim, a mesma promove um equilíbrio das duas métricas, se uma das duas for ínfima o *f-score* refletirá essa disparidade.

$$\textit{f-score} = 2 * \frac{precision * recall}{precision + recall} \tag{25}$$

Foi usado também para extrair as médias dos resultados sobre qualquer métrica desejada em dois tipos, médias macro e ponderada.

A média macro ou harmônica. É a razão da somatória da medidas por quantidade total de medidas, de forma que a_n descreve o valor da medida n-ésima, no qual o índice é n medida e N o valor da quantidade de medidas total (Equação 26):

$$\textit{Média Macro} = \frac{a_1 + a_2 + ... + a_n}{N} \tag{26}$$

E para avaliar a métrica com relação dos valores por significância de classe, em que cada valor de amostra a_n é multiplicado por um peso definido por quanto representativa é a amostra em relação a quantidade total de amostras, assim descrito na Equação 27.

$$\textit{Média Ponderada} = \frac{(a_1 * p_{classe}) + (a_2 * p_{classe}) + ... + (a_n * p_{classe})}{N} \tag{27}$$

3.2 EXPERIMENTOS

3.2.1 Descrição do *dataset* de Vídeos

Para a realizacão dos experimentos foi utitilizado o *dataset* público de vídeos denominado UCF101 (SOOMRO *et al.*, 2012), sendo que o mesmo é um dos aplicados em trabalhos

do estado da arte. O *dataset* foi escolhido por sofrer uma alteração pequena em volume de amostras por classe e um cenário real de ambientes com uma quantidade de 13320 vídeos. Esse conjunto de dados é composto por vídeos da web que são gravados em ambientes irrestritos com variações relacionadas à movimentação de camêra, condições de luminosidade, oclusões e *frames* com baixa qualidade. A denominação UCF101 refere-se à quantidade de classes do conjunto (i.e., 101 classes). O mesmo é composto por vídeos de ações envolvendo entes humanos e/ou objetos. Essas interações são divididas em 5 categorias globais, sendo interação humano-objeto, humano-humano, somente corpo em movimento, tocar instrumentos musicais e esportes. A Figura 15 ilustra uma amostra de *frame* de cada uma das 101 classes do dataset. Um exemplo de tais classes são: *Apply Eye Makeup, Apply Lipstick, Archery, Baby Crawling, Balance Beam, Band Marching, Baseball Pitch, Basketball Shooting, Basketball Dunk, Bench Press, Biking,* entre outras. A Tabela 2 ilustra a distribuição de vídeos para cada uma das classes.

Com relação às especificações do *dataset* UCF101, a Tabela 3 descreve as mesmas. É possível verificar que as amostras (vídeos) variam suas quantidades de frames, tendo uma média de 180 *frames* baseada na média de duração e na taxa de atualização (*frame rate*) dos vídeos. Para os experimentos, foi utilizada uma *graphics processing unit* (GPU) a qual limitou-se a carregar 39.960 imagens como *subset* total, tanto para teste quanto para treino. Essa foi uma restrição computacional estabelecida para os experimentos, dadas as possiblidades de *hardware* disponibilizadas para o desenvolvimento do presente trabalho. Assim, em questões de divisão do total de vídeos é o equivalente a $f = 3$ (i.e., foi possível carregar 3 *frames* de cada vídeo dada a cardinalidade do *dataset* agregadas às limitações computacionais do momento).

Dessa forma, os *frames* escolhidos foram retirados de cada um dos 13320 vídeos aplicando a política de seleção de *frames* explicitadas anteriormente (ver Seção 3.1.1). Por exemplo, obtendo assim um banco de imagens com cardinaldade igual a $13320 * f$. Além disso, foram aplicadas as duas estratégias de conexão e definição de nós explicitadas na Seção 3.1.3.

3.2.2 Cenário Experimental

As características visuais de cada *frame* foram extraídas utilizando características profundas (*deep features*) oriundas da arquitetura *Resnet-101* (HE *et al.*, 2015), com aplicação do conceito *transfer learning* com pesos pré-treinados pelo *ImageNet* dataset. Apesar da arquitutera *Resnet-101* ter sido utilizada qualquer outra arquitetura pode ser utilizada na metologia proposta. Optou-se por tal arquitetura pois ela apresenta bom *trade-off* relacionado a eficiência e eficácia.

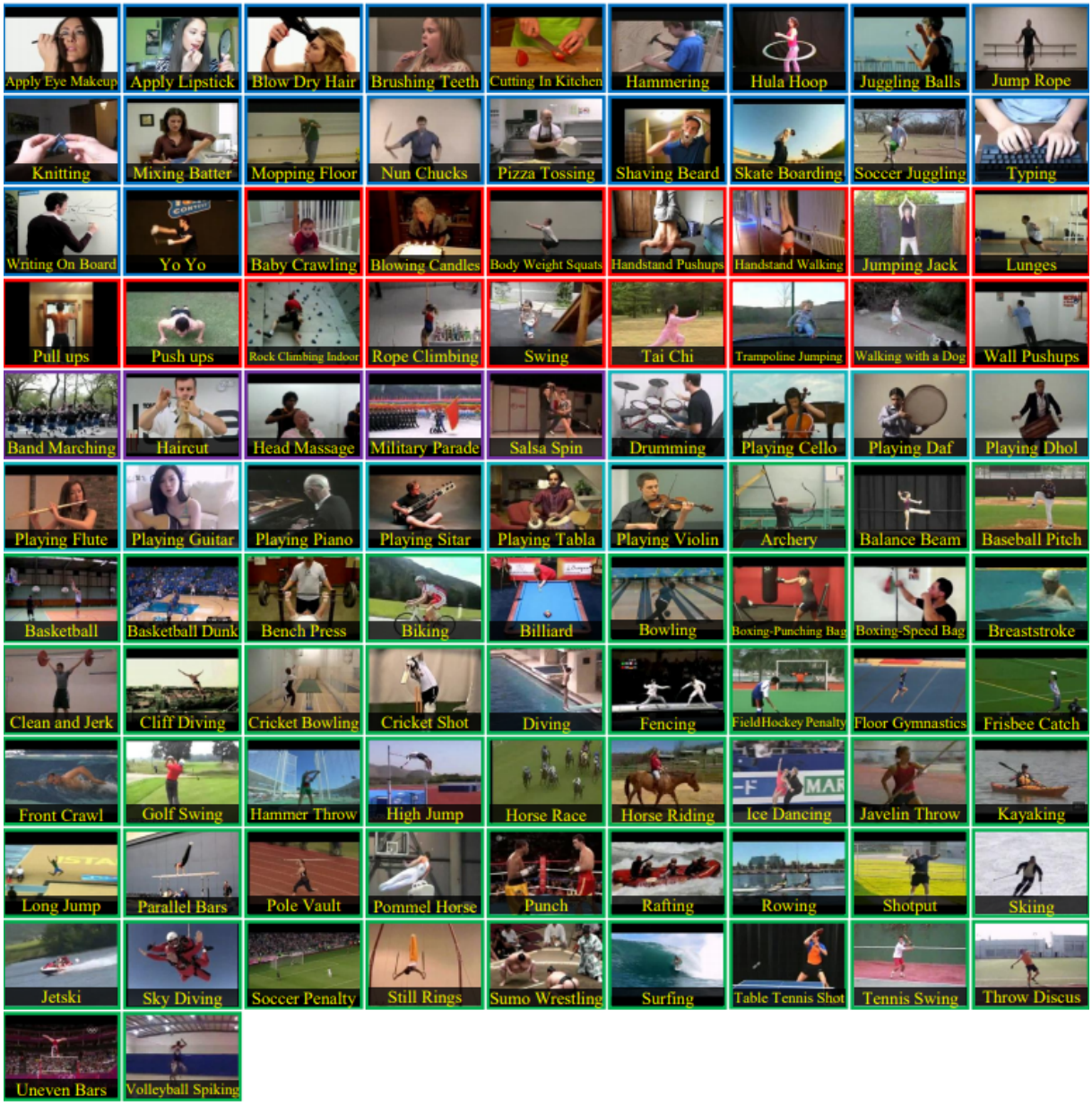

Figura 15 – Amostra das 101 classes do *dataset* UCF101.
Fonte: (SOOMRO *et al.*, 2012)

Com relação aos conjuntos de treinamento e teste utilizados foi estipulado 75% do *dataset* para treinamento e 25% para teste. Tais conjuntos foram criados mantendo-se a proporcionalidade interna de cada uma das classes. Por fim, o otimizador utilizado tanto para a etapa de extração de características por meio das *deep features*, bem como para o emprego da GCN foi o Adam (KINGMA; BA, 2014).

A metodologia foi comparada a três modelos de redes *end-to-end* baseados em arquitetura convolucional. Dois dos modelos foram propostos em (TRAN *et al.*, 2014), que trata-se do método denominado C3D e uma variação homogênea do mesmo. O terceiro modelo da literatura comparado utiliza filtros convolucionais para extração de características vinculados a células

Tabela 2 – Descrição de Classes UCF101

Índice	Classe	Qtd. Amostras	Índice	Classe	Qtd. Amostras
1	Apply EyeMakeup	44	52	Lunges	37
2	Apply Lipstick	27	53	Military Parade	33
3	Archery	38	54	Mixing	45
4	Baby Crawling	34	55	Mopping Floor	31
5	Balance Beam	31	56	Nunchucks	34
6	Band Marching	38	57	Parallel Bars	37
7	Baseball Pitch	43	58	Pizza Tossing	33
8	Basketball	35	59	Playing Cello	41
9	Basketball Dunk	37	60	Playing Daf	16
10	Bench Press	48	61	Playing Dhol	26
11	Biking	32	62	Playing Flute	37
12	Billiards	20	63	Playing Guitar	43
13	Blow DryHair	37	64	Playing Piano	28
14	Blowing Candles	33	65	Playing Sitar	18
15	Body WeightSquats	30	66	Playing Tabla	24
16	Bowling	43	67	Playing Violin	26
17	Boxing PunchingBag	48	68	Pole Vault	38
18	Boxing SpeedBag	30	69	Pommel Horse	12
19	Breast Stroke	25	70	Pull Ups	28
20	Brushing Teeth	27	71	Punch	39
21	Clean And Jerk	32	72	Push Ups	30
22	Cliff Diving	39	73	Rafting	28
23	Cricket Bowling	36	74	Rock Climbing Indoor	6
24	Cricket Shot	49	75	Rope Climbing	34
25	CuttingIn Kitchen	32	76	Rowing	11
26	Diving	45	77	Salsa Spin	30
27	Drumming	45	78	Shaving Beard	32
28	Fencing	34	79	Shotput	46
29	FieldHockey Penalty	40	80	Skate Boarding	31
30	Floor Gymnastics	36	81	Skiing	39
31	Frisbee Catch	37	82	Skijet	20
32	Front Crawl	32	83	Sky Diving	31
33	Golf Swing	36	84	Soccer Juggling	22
34	Haircut	33	85	Soccer Penalty	41
35	Hammering	33	86	Still Rings	32
36	Hammer Throw	45	87	Sumo Wrestling	33
37	Handstand Pushups	28	88	Surfing	33
38	Handstand Walking	34	89	Swing	40
39	Head Massage	32	90	Table TennisShot	39
40	High Jump	37	91	Tai Chi	28
41	Horse Race	33	92	Tennis Swing	49
42	Horse Riding	49	93	Throw Discus	38
43	Hula Hoop	34	94	Trampoline Jumping	28
44	Ice Dancing	46	95	Typing	43
45	Javelin Throw	31	96	Uneven Bars	28
46	Juggling Balls	40	97	Volleyball Spiking	35
47	Jumping Jack	37	98	Walking With Dog	36
48	Jump Rope	5	99	Wall Pushups	35
49	Kayaking	36	100	Writing On Board	45
50	Knitting	34	101	YoYo	36
51	Long Jump	39			

Tabela 3 – Informações do Dataset

Ações	101
Vídeos	13320
Grupos por Ações	25
Vídeos por Grupo	4-7
Média de Duração dos Vídeos	7.21 seg
Total de Duração dos Vídeos	1600 min
Duração Mínima de Vídeo	1.06 seg
Duração Máxima de Vídeo	71.04 seg
Taxa de Quadros	25 fps
Resolução	320x240
Áudio	Sim (51 Ações)

Fonte: (SOOMRO *et al.*, 2012)

LSTM (DONAHUE *et al.*, 2014). Como foi executado com recursos limitados, cada modelo teve um subconjunto do *dataset* ajustado para as dimensões das redes. A Tabela 4 apresenta a parametrização considerada para cada um dos métodos utilizados nos experimentos.

Tabela 4 – Parametrização das arquiteturas utilizadas nos experimentos.

Abordagem	Batch Size	*frames*	Learning Rate	Épocas	Otimizador	Early Stop
C3D	2	16	0.05	1000	Adam	50
LCRN	76	10	0.05	1000	Adam	50
Conv 3D	46	10	0.05	1000	Adam	50
GCN - Sequencial (Proposta)	39960	3	0.01	15000	Adam	50
GCN - Limiar (Proposta)	13320	3	0.05	4000	Adam	50

3.2.3 Resultados

Inicialmente fora realizados experimentos considerando a primeira estratégia de conexão com a GCN utilizando *frames* individuais (cada *frame* caracteriza-se por um nó no grafo) para classificação (ver abordagem descrita na Figura 14). Em tais experimentos, foram obtidos pelo método proposto os resultados explicitados na Tabela 5, a mesma apresenta as métricas de precisão, revocação, *f-score* e acurácia para cada uma das classes do *dataset* de teste utilizado. Além disso, foi também gerada a matriz de confusão dos mesmos ilustrada na Figura 16, a qual foi ponderada por meio de um mapa de calor para melhor visualizacão. Nesse referido experimento foi fixado o valor de 15000 épocas e uma taxa de aprendizado de 0.01 (ver Tabela 4 da seção 3.2.2), a qual é relativamente baixa em relação aos outros métodos. A escolha da quantidade grande de épocas para o método proposto é devido à taxa de aprendizado inicial, agregado ao fato do mesmo executar todo o lote de amostras em uma única época. Além disso, ao aplicar taxas de aprendizado maiores o modelo de aprendizado apresentou maior instabilidade.

Ao comparar os resultados obtidos pelo método proposto empregando a primeira estratégia (sequencial), com os obtidos pelos da literatura (i.e., modelos baseados em CNN) obteve-se um ganho significativo chegando quase ao dobro de acurácia do modelo C3D. Vale ressaltar que o C3D, ao ser comparado com o Conv3D e o LRCN (métodos da literatura) foi o mais bem sucedido. Os resultados obtidos pelos métodos da literatura C3D, Conv3d e LRCN estão explicitados nas Tabelas 6 a 8, respectivamente. Além disso, as matrizes de confusão dos mesmos visualizadas por meio do mapa de calor são apresentadas nas Figuras 7 a 19, respectivamente.

Apesar de ter obtido resultados satisfatórios em termos de eficácia, o método proposto aplicando a primeira estratégia apresenta-se de certa forma oneroso pois necessitou de várias

épocas. Além disso, um outro ponto a ser citado é que ao compará-lo a outros modelos da literatura o método não é totalmente concordante para predições considerando cada amostra como um vídeo, visto que ele considera cada *frame* como uma amostra. Assim, o mesmo limita-se à predição de amostras como *frames* individuais onde cada um pode ser classificado como classes diferentes, mesmo pertencendo ao mesmo vídeo.

Com o intuito de mitigar os pontos elencados e tornar a comparação mais adequada em relação aos métodos da literatura, foram também realizados experimentos considerando a segunda estratégia de conexão dos nós aplicada pelo método proposto, a qual considera limiares como critérios de poda das conexões. Para tanto, foram definidos 3 limiares diferentes sendo os mesmos 0.15, 0.25 e 0.5, respectivamente. Esses limiares foram estabelecidos empiricamente no presente trabalho e visaram podar conexões de acordo com diferentes gradações de similaridade entre os nós do grafo, inicialmente considerando gradações maiores de similaridade e diminuindo a mesma de maneira escalonada. Dessa forma, o objetivo foi também analisar o comportamento das podas em relação aos diferentes níveis de similaridade impostos pelos limiares considerados.

Ao comparar o método proposto utilizando a segunda estratégida de conexão dos nós em relação aos outros da literatura, novamente pode-se perceber um ganho expressivo em relação a todas as métricas consideradas. As Tabelas 9, 10 e 11 explicitam tais resultados obtidos considerando os limiares de 0.15, 0.25 e 0.5, respectivamente. Os resultados de acurácia geral foram todos acima de 90%, com os limiares de 0.25 e 0.50 apresentando um empate na acurácia média obtida. A única exceção de resultados inferiores foi referente à classe *Jump Rope*, o qual se repetiu em todos os experimentos de forma unânime.

Um possível motivo dos resultados obtidos serem próximos, para a segunda estratégia, são os limiares serem muito inferiores à média das distâncias entre todos os vídeos. Assim, esse fato recai sobre um problema similar ao de definição de raio de abrangência, pois para obter um limiar próximo do ideal seria adequado obter o diâmetro do espaço de busca. Outra possbilidade seria extrair informações baseadas em medidas de grafos para melhor definição do limiar. No entanto, tais análises não fizeram parte do escopo do presente trabalho.

Além de tais resultados, foram também geradas as matrizes de confusão dos resultados considerandos os respectivos limiares, as quais são ilustradas nas Figuras 20 a 22. Ao comparar tais visualizações das matrizes em relação às obtidas pelos métodos da literatura (Figuras 7 a 19) pode-se perceber a superioridade do método proposto em relação às mesmas, visto que a frequência do mesmo apresenta uma concentração muito maior na diagonal principal da matriz.

Por fim, a Tabela 12 sumariza as todas as métricas globais obtidas nos experimentos realizados, considerando os métodos da literatura e o método proposto com suas diferentes estratégias. Assim, pode-se perceber que enquanto o C3D obteve uma acurácia global de 33%, o método proposto sequencial (primeira estratégia) obteve 62%. Já, ao aplicar a segunda estratégia foi obtida uma acurácia de até 92%. Comportamento similar foi observado para as outras métricas consideradas.

Tabela 5 – Resultados GCN - Sequencial

GCN - Sequencial									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.74	0.77	0.86	0.76	Lunges	0.40	0.22	0.72	0.28
Apply Lipstick	0.70	0.59	0.89	0.64	Military Parade	0.88	0.70	0.96	0.78
Archery	0.57	0.68	0.88	0.62	Mixing	0.86	0.80	1.00	0.83
Baby Crawling	0.58	0.76	0.97	0.66	Mopping Floor	0.53	0.26	0.89	0.35
Balance Beam	0.83	0.32	0.89	0.47	Nunchucks	0.13	0.06	0.75	0.08
Band Marching	0.63	0.95	0.94	0.76	Parallel Bars	0.61	0.68	0.93	0.64
Baseball Pitch	0.66	0.67	0.99	0.67	Pizza Tossing	0.44	0.21	0.84	0.29
Basketball	0.33	0.34	0.97	0.34	Playing Cello	0.89	0.80	0.99	0.85
Basketball Dunk	0.71	1.00	0.80	0.83	Playing Daf	0.80	1.00	0.98	0.89
Bench Press	0.79	0.94	0.96	0.86	Playing Dhol	0.69	0.85	1.00	0.76
Biking	0.74	0.91	0.96	0.82	Playing Flute	0.88	0.76	0.98	0.81
Billiards	0.91	1.00	1.00	0.95	Playing Guitar	0.75	1.00	0.98	0.86
Blow DryHair	0.67	0.32	0.90	0.44	Playing Piano	0.96	0.89	0.99	0.93
Blowing Candles	0.76	0.85	0.96	0.80	Playing Sitar	0.80	0.22	1.00	0.35
Body WeightSquats	0.43	0.20	0.88	0.27	Playing Tabla	0.84	0.67	0.95	0.74
Bowling	0.83	0.93	0.98	0.88	Playing Violin	0.92	0.85	1.00	0.88
Boxing PunchingBag	0.52	0.48	0.95	0.50	Pole Vault	0.76	0.76	0.90	0.76
Boxing SpeedBag	0.63	0.57	0.89	0.60	Pommel Horse	0.83	0.42	0.80	0.56
Breast Stroke	0.65	0.60	0.88	0.63	Pull Ups	0.57	0.29	0.90	0.38
Brushing Teeth	0.32	0.22	0.85	0.26	Punch	0.64	0.87	0.95	0.74
Clean And Jerk	0.71	0.47	0.97	0.57	Push Ups	0.71	0.57	0.95	0.63
Cliff Diving	0.68	0.64	0.89	0.66	Rafting	0.59	0.86	0.98	0.70
Cricket Bowling	0.38	0.47	0.80	0.42	Rock Climbing Indoor	0.80	0.67	0.89	0.73
Cricket Shot	0.31	0.35	0.89	0.33	Rope Climbing	0.64	0.41	0.79	0.50
CuttingIn Kitchen	0.78	0.88	1.00	0.82	Rowing	0.47	0.73	0.95	0.57
Diving	0.70	0.96	0.98	0.81	Salsa Spin	0.53	0.33	0.90	0.41
Drumming	0.71	0.76	0.99	0.73	Shaving Beard	0.41	0.75	0.88	0.53
Fencing	0.72	0.76	0.91	0.74	Shotput	0.40	0.43	0.82	0.42
FieldHockey Penalty	0.22	0.17	0.85	0.19	Skate Boarding	0.62	0.52	0.89	0.56
Floor Gymnastics	0.62	0.44	0.91	0.52	Skiing	0.66	0.79	0.96	0.72
Frisbee Catch	0.51	0.62	0.89	0.56	Skijet	0.81	0.85	0.94	0.83
Front Crawl	0.65	0.75	0.91	0.70	Sky Diving	0.71	0.94	0.92	0.81
Golf Swing	0.55	0.47	0.86	0.51	Soccer Juggling	0.21	0.27	0.73	0.24
Haircut	0.42	0.67	0.93	0.51	Soccer Penalty	0.74	0.83	0.96	0.78
Hammering	0.48	0.42	0.96	0.45	Still Rings	0.81	0.69	0.96	0.75
Hammer Throw	0.47	0.62	0.88	0.54	Sumo Wrestling	0.83	0.73	1.00	0.77
Handstand Pushups	0.28	0.39	0.97	0.32	Surfing	0.74	0.94	0.94	0.83
Handstand Walking	0.00	0.00	0.70	0.00	Swing	0.74	0.65	0.94	0.69
Head Massage	0.43	0.78	0.96	0.56	Table TennisShot	0.93	1.00	0.98	0.96
High Jump	0.50	0.24	0.81	0.33	Tai Chi	0.37	0.46	0.92	0.41
Horse Race	0.74	0.70	0.97	0.72	Tennis Swing	0.28	0.18	0.92	0.22
Horse Riding	0.77	0.98	0.99	0.86	Throw Discus	0.45	0.74	0.77	0.56
Hula Hoop	0.43	0.35	0.78	0.39	Trampoline Jumping	0.43	0.75	0.90	0.55
Ice Dancing	0.80	0.98	1.00	0.88	Typing	0.93	0.91	1.00	0.92
Javelin Throw	0.00	0.00	0.77	0.00	Uneven Bars	0.55	0.75	0.94	0.64
Juggling Balls	0.62	0.53	0.95	0.57	Volleyball Spiking	0.48	0.63	0.85	0.54
Jumping Jack	0.16	0.11	0.93	0.13	Walking With Dog	0.67	0.56	0.83	0.61
Jump Rope	0.00	0.00	0.00	0.00	Wall Pushups	0.36	0.26	0.92	0.30
Kayaking	0.75	0.42	0.98	0.54	Writing On Board	0.69	0.84	0.98	0.76
Knitting	0.94	0.97	1.00	0.96	YoYo	0.46	0.50	0.95	0.48
Long Jump	0.57	0.44	0.93	0.49					

Tabela 6 – Resultados Conv 3D

Conv 3D									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.28	0.16	0.86	0.20	Lunges	0.05	0.08	0.72	0.06
Apply Lipstick	0.33	0.33	0.89	0.33	Military Parade	0.19	0.18	0.96	0.18
Archery	0.00	0.00	0.88	0.00	Mixing	0.40	0.40	1.00	0.40
Baby Crawling	0.12	0.12	0.97	0.12	Mopping Floor	0.13	0.10	0.89	0.11
Balance Beam	0.18	0.06	0.89	0.10	Nunchucks	0.03	0.03	0.75	0.03
Band Marching	0.42	0.63	0.94	0.51	Parallel Bars	0.47	0.22	0.93	0.30
Baseball Pitch	0.51	0.51	0.99	0.51	Pizza Tossing	0.25	0.06	0.84	0.10
Basketball	0.24	0.29	0.97	0.26	Playing Cello	0.00	0.00	0.99	0.00
Basketball Dunk	0.56	0.81	0.80	0.66	Playing Daf	0.00	0.00	0.98	0.00
Bench Press	0.34	0.56	0.96	0.43	Playing Dhol	0.07	0.12	1.00	0.08
Biking	0.25	0.19	0.96	0.21	Playing Flute	0.09	0.05	0.98	0.07
Billiards	0.86	0.95	1.00	0.90	Playing Guitar	0.47	0.44	0.98	0.46
Blow DryHair	0.27	0.11	0.90	0.15	Playing Piano	0.49	0.64	0.99	0.55
Blowing Candles	0.26	0.27	0.96	0.27	Playing Sitar	0.00	0.00	1.00	0.00
Body WeightSquats	0.12	0.13	0.88	0.13	Playing Tabla	0.29	0.25	0.95	0.27
Bowling	0.42	0.70	0.98	0.53	Playing Violin	0.33	0.27	1.00	0.30
Boxing PunchingBag	0.20	0.19	0.95	0.19	Pole Vault	0.31	0.47	0.90	0.37
Boxing SpeedBag	0.00	0.00	0.89	0.00	Pommel Horse	0.83	0.42	0.80	0.56
Breast Stroke	0.40	0.76	0.88	0.52	Pull Ups	0.15	0.11	0.90	0.12
Brushing Teeth	0.11	0.15	0.85	0.13	Punch	0.54	0.77	0.95	0.63
Clean And Jerk	0.33	0.09	0.97	0.15	Push Ups	0.00	0.00	0.95	0.00
Cliff Diving	0.47	0.36	0.89	0.41	Rafting	0.33	0.36	0.98	0.34
Cricket Bowling	0.20	0.47	0.80	0.28	Rock Climbing Indoor	0.00	0.00	0.89	0.00
Cricket Shot	0.29	0.45	0.89	0.35	Rope Climbing	0.36	0.15	0.79	0.21
CuttingIn Kitchen	0.44	0.25	1.00	0.32	Rowing	0.13	0.18	0.95	0.15
Diving	0.37	0.49	0.98	0.42	Salsa Spin	0.46	0.20	0.90	0.28
Drumming	0.16	0.16	0.99	0.16	Shaving Beard	0.07	0.06	0.88	0.07
Fencing	0.26	0.21	0.91	0.23	Shotput	0.03	0.02	0.82	0.03
FieldHockey Penalty	0.52	0.42	0.85	0.47	Skate Boarding	0.23	0.23	0.89	0.23
Floor Gymnastics	0.19	0.17	0.91	0.18	Skiing	0.44	0.46	0.96	0.45
Frisbee Catch	0.31	0.57	0.89	0.40	Skijet	0.30	0.30	0.94	0.30
Front Crawl	0.46	0.38	0.91	0.41	Sky Diving	0.35	0.39	0.92	0.37
Golf Swing	0.38	0.25	0.86	0.30	Soccer Juggling	0.00	0.00	0.73	0.00
Haircut	0.02	0.03	0.93	0.03	Soccer Penalty	0.73	0.98	0.96	0.83
Hammering	0.00	0.00	0.96	0.00	Still Rings	0.79	0.34	0.96	0.48
Hammer Throw	0.45	0.47	0.88	0.46	Sumo Wrestling	0.43	0.36	1.00	0.39
Handstand Pushups	0.08	0.14	0.97	0.10	Surfing	0.61	0.58	0.94	0.59
Handstand Walking	0.00	0.00	0.70	0.00	Swing	0.24	0.40	0.94	0.30
Head Massage	0.00	0.00	0.96	0.00	Table TennisShot	0.33	0.26	0.98	0.29
High Jump	0.27	0.16	0.81	0.20	Tai Chi	0.14	0.14	0.92	0.14
Horse Race	0.59	0.52	0.97	0.55	Tennis Swing	0.22	0.18	0.92	0.20
Horse Riding	0.36	0.71	0.99	0.48	Throw Discus	0.14	0.13	0.77	0.14
Hula Hoop	0.33	0.24	0.78	0.28	Trampoline Jumping	0.29	0.50	0.90	0.36
Ice Dancing	0.60	0.80	1.00	0.69	Typing	0.05	0.02	1.00	0.03
Javelin Throw	0.40	0.26	0.77	0.31	Uneven Bars	0.28	0.61	0.94	0.39
Juggling Balls	0.29	0.17	0.95	0.22	Volleyball Spiking	0.33	0.29	0.85	0.31
Jumping Jack	0.47	0.43	0.93	0.45	Walking With Dog	0.14	0.08	0.83	0.10
Jump Rope	0.00	0.00	0.00	0.00	Wall Pushups	0.11	0.17	0.92	0.13
Kayaking	0.21	0.19	0.98	0.20	Writing On Board	0.33	0.31	0.98	0.32
Knitting	0.19	0.15	1.00	0.17	YoYo	0.13	0.11	0.95	0.12
Long Jump	0.29	0.28	0.93	0.29					

Tabela 7 – Resultados C3D

C3D									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.37	0.39	0.38	38.64	Lunges	0.33	0.30	0.31	29.73
Apply Lipstick	0.16	0.19	0.17	18.52	Military Parade	0.19	0.27	0.22	27.27
Archery	0.04	0.03	0.03	2.63	Mixing	0.24	0.36	0.29	35.56
Baby Crawling	0.21	0.18	0.19	17.65	Mopping Floor	0.20	0.23	0.21	22.58
Balance Beam	0.30	0.23	0.26	22.58	Nunchucks	0.00	0.00	0.00	0.0
Band Marching	0.41	0.58	0.48	57.89	Parallel Bars	0.46	0.73	0.56	72.97
Baseball Pitch	0.56	0.63	0.59	62.79	Pizza Tossing	0.00	0.00	0.00	0.0
Basketball	0.29	0.46	0.36	45.71	Playing Cello	0.00	0.00	0.00	0.0
Basketball Dunk	0.55	0.92	0.69	91.89	Playing Daf	0.00	0.00	0.00	0.0
Bench Press	0.31	0.40	0.35	39.58	Playing Dhol	0.00	0.00	0.00	0.0
Biking	0.47	0.53	0.50	53.12	Playing Flute	0.00	0.00	0.00	0.0
Billiards	0.79	0.95	0.86	95.0	Playing Guitar	0.65	0.40	0.49	39.53
Blow DryHair	0.30	0.22	0.25	21.62	Playing Piano	0.42	0.86	0.56	85.71
Blowing Candles	0.21	0.36	0.27	36.36	Playing Sitar	0.01	0.06	0.02	5.56
Body WeightSquats	0.15	0.17	0.16	16.67	Playing Tabla	0.12	0.21	0.15	20.83
Bowling	0.55	0.65	0.60	65.12	Playing Violin	0.22	0.27	0.24	26.92
Boxing PunchingBag	0.00	0.00	0.00	0.0	Pole Vault	0.53	0.21	0.30	21.05
Boxing SpeedBag	0.00	0.00	0.00	0.0	Pommel Horse	0.57	0.33	0.42	33.33
Breast Stroke	0.73	0.96	0.83	96.0	Pull Ups	0.00	0.00	0.00	0.0
Brushing Teeth	0.00	0.00	0.00	0.0	Punch	0.74	0.64	0.68	64.1
Clean And Jerk	0.23	0.31	0.26	31.25	Push Ups	0.00	0.00	0.00	0.0
Cliff Diving	0.39	0.28	0.33	28.21	Rafting	0.57	0.61	0.59	60.71
Cricket Bowling	0.32	0.67	0.44	66.67	Rock Climbing Indoor	0.00	0.00	0.00	0.0
Cricket Shot	0.49	0.35	0.40	34.69	Rope Climbing	0.12	0.06	0.08	5.88
CuttingIn Kitchen	0.30	0.09	0.14	9.38	Rowing	0.06	0.09	0.07	9.09
Diving	0.70	0.71	0.70	71.11	Salsa Spin	0.00	0.00	0.00	0.0
Drumming	0.11	0.04	0.06	4.44	Shaving Beard	0.29	0.12	0.17	12.5
Fencing	0.21	0.09	0.12	8.82	Shotput	0.18	0.15	0.16	15.22
FieldHockey Penalty	0.46	0.70	0.55	70.0	Skate Boarding	0.27	0.29	0.28	29.03
Floor Gymnastics	0.45	0.53	0.49	52.78	Skiing	0.62	0.33	0.43	33.33
Frisbee Catch	0.32	0.51	0.40	51.35	Skijet	0.70	0.35	0.47	35.0
Front Crawl	0.76	0.41	0.53	40.62	Sky Diving	0.33	0.35	0.34	35.48
Golf Swing	0.33	0.58	0.42	58.33	Soccer Juggling	0.10	0.14	0.11	13.64
Haircut	0.00	0.00	0.00	0.0	Soccer Penalty	0.76	0.83	0.79	82.93
Hammering	0.00	0.00	0.00	0.0	Still Rings	0.46	0.38	0.41	37.5
Hammer Throw	0.36	0.33	0.34	33.33	Sumo Wrestling	0.40	0.52	0.45	51.52
Handstand Pushups	0.18	0.07	0.10	7.14	Surfing	1.00	0.36	0.53	36.36
Handstand Walking	0.20	0.06	0.09	5.88	Swing	0.47	0.40	0.43	40.0
Head Massage	0.25	0.12	0.17	12.5	Table TennisShot	0.12	0.03	0.04	2.56
High Jump	0.54	0.19	0.28	18.92	Tai Chi	0.24	0.39	0.30	39.29
Horse Race	0.78	0.42	0.55	42.42	Tennis Swing	0.67	0.29	0.40	28.57
Horse Riding	0.57	0.73	0.64	73.47	Throw Discus	0.16	0.21	0.18	21.05
Hula Hoop	0.15	0.29	0.20	29.41	Trampoline Jumping	0.26	0.61	0.36	60.71
Ice Dancing	0.79	0.98	0.87	97.83	Typing	0.52	0.28	0.36	27.91
Javelin Throw	0.50	0.42	0.46	41.94	Uneven Bars	0.61	0.79	0.69	78.57
Juggling Balls	0.07	0.03	0.04	2.5	Volleyball Spiking	0.40	0.80	0.53	80.0
Jumping Jack	0.64	0.19	0.29	18.92	Walking With Dog	0.60	0.08	0.15	8.33
Jump Rope	0.00	0.00	0.00	0.0	Wall Pushups	0.08	0.17	0.11	17.14
Kayaking	0.36	0.22	0.28	22.22	Writing On Board	0.35	0.29	0.32	28.89
Knitting	0.68	0.50	0.58	50.0	YoYo	0.12	0.08	0.10	8.33
Long Jump	0.21	0.21	0.21	20.51					

Tabela 8 – Resultados LRCN

LRCN									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.00	0.00	0.00	0.0	Lunges	0.06	0.05	0.06	5.41
Apply Lipstick	0.20	0.26	0.23	25.93	Military Parade	0.13	0.21	0.16	21.21
Archery	0.16	0.11	0.13	10.53	Mixing	0.26	0.29	0.27	28.89
Baby Crawling	0.10	0.18	0.12	17.65	Mopping Floor	0.08	0.03	0.05	3.23
Balance Beam	0.16	0.10	0.12	9.68	Nunchucks	0.00	0.00	0.00	0.0
Band Marching	0.25	0.37	0.30	36.84	Parallel Bars	0.35	0.22	0.27	21.62
Baseball Pitch	0.44	0.53	0.48	53.49	Pizza Tossing	0.00	0.00	0.00	0.0
Basketball	0.23	0.40	0.29	40.0	Playing Cello	0.10	0.10	0.10	9.76
Basketball Dunk	0.46	0.86	0.60	86.49	Playing Daf	0.00	0.00	0.00	0.0
Bench Press	0.17	0.38	0.24	37.5	Playing Dhol	0.00	0.00	0.00	0.0
Biking	0.12	0.09	0.10	9.38	Playing Flute	0.00	0.00	0.00	0.0
Billiards	1.00	0.95	0.97	95.0	Playing Guitar	0.55	0.37	0.44	37.21
Blow DryHair	0.15	0.14	0.14	13.51	Playing Piano	0.47	0.64	0.55	64.29
Blowing Candles	0.23	0.42	0.30	42.42	Playing Sitar	0.00	0.00	0.00	0.0
Body WeightSquats	0.10	0.10	0.10	10.0	Playing Tabla	0.00	0.00	0.00	0.0
Bowling	0.37	0.51	0.43	51.16	Playing Violin	0.25	0.31	0.28	30.77
Boxing PunchingBag	0.12	0.19	0.15	18.75	Pole Vault	0.29	0.53	0.37	52.63
Boxing SpeedBag	0.06	0.03	0.04	3.33	Pommel Horse	0.00	0.00	0.00	0.0
Breast Stroke	0.79	0.88	0.83	88.0	Pull Ups	0.07	0.04	0.05	3.57
Brushing Teeth	0.00	0.00	0.00	0.0	Punch	0.35	0.69	0.46	69.23
Clean And Jerk	0.50	0.09	0.16	9.38	Push Ups	0.00	0.00	0.00	0.0
Cliff Diving	0.30	0.21	0.24	20.51	Rafting	0.21	0.21	0.21	21.43
Cricket Bowling	0.28	0.31	0.29	30.56	Rock Climbing Indoor	0.00	0.00	0.00	0.0
Cricket Shot	0.06	0.06	0.06	6.12	Rope Climbing	0.08	0.03	0.04	2.94
CuttingIn Kitchen	0.08	0.03	0.04	3.12	Rowing	0.29	0.18	0.22	18.18
Diving	0.34	0.58	0.43	57.78	Salsa Spin	0.00	0.00	0.00	0.0
Drumming	0.00	0.00	0.00	0.0	Shaving Beard	0.03	0.03	0.03	3.12
Fencing	0.57	0.12	0.20	11.76	Shotput	0.05	0.02	0.03	2.17
FieldHockey Penalty	0.33	0.28	0.30	27.5	Skate Boarding	0.20	0.29	0.24	29.03
Floor Gymnastics	0.14	0.06	0.08	5.56	Skiing	0.32	0.31	0.31	30.77
Frisbee Catch	0.23	0.35	0.28	35.14	Skijet	0.30	0.50	0.38	50.0
Front Crawl	0.42	0.47	0.44	46.88	Sky Diving	0.35	0.42	0.38	41.94
Golf Swing	0.11	0.22	0.15	22.22	Soccer Juggling	0.00	0.00	0.00	0.0
Haircut	0.09	0.15	0.11	15.15	Soccer Penalty	0.56	0.83	0.67	82.93
Hammering	0.06	0.03	0.04	3.03	Still Rings	0.35	0.38	0.36	37.5
Hammer Throw	0.28	0.29	0.28	28.89	Sumo Wrestling	0.15	0.12	0.13	12.12
Handstand Pushups	0.00	0.00	0.00	0.0	Surfing	0.34	0.76	0.47	75.76
Handstand Walking	0.20	0.03	0.05	2.94	Swing	0.13	0.10	0.11	10.0
Head Massage	0.00	0.00	0.00	0.0	Table TennisShot	0.09	0.03	0.04	2.56
High Jump	0.15	0.05	0.08	5.41	Tai Chi	0.00	0.00	0.00	0.0
Horse Race	0.39	0.48	0.43	48.48	Tennis Swing	0.22	0.27	0.24	26.53
Horse Riding	0.21	0.49	0.29	48.98	Throw Discus	0.23	0.13	0.17	13.16
Hula Hoop	0.26	0.35	0.30	35.29	Trampoline Jumping	0.09	0.25	0.13	25.0
Ice Dancing	0.44	0.89	0.59	89.13	Typing	0.00	0.00	0.00	0.0
Javelin Throw	0.00	0.00	0.00	0.0	Uneven Bars	0.41	0.57	0.48	57.14
Juggling Balls	0.09	0.03	0.04	2.5	Volleyball Spiking	0.47	0.46	0.46	45.71
Jumping Jack	0.62	0.22	0.32	21.62	Walking With Dog	0.07	0.03	0.04	2.78
Jump Rope	0.00	0.00	0.00	0.0	Wall Pushups	0.27	0.17	0.21	17.14
Kayaking	0.14	0.11	0.12	11.11	Writing On Board	0.16	0.16	0.16	15.56
Knitting	0.29	0.47	0.36	47.06	YoYo	0.05	0.03	0.04	2.78
Long Jump	0.21	0.18	0.19	17.95					

Tabela 9 – Resultados GCN - Limiar = 0.15

GCN - Limiar = 0.15									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.81	0.94	0.87	94.44	Lunges	0.65	0.79	0.71	78.95
Apply Lipstick	0.91	0.79	0.85	78.95	Military Parade	0.90	0.93	0.91	92.5
Archery	0.88	0.86	0.87	85.71	Mixing	0.86	1.00	0.93	100.0
Baby Crawling	0.91	0.98	0.94	97.67	Mopping Floor	0.85	0.74	0.79	73.91
Balance Beam	0.77	0.84	0.81	84.38	Nunchucks	0.79	0.66	0.72	65.71
Band Marching	0.86	0.89	0.87	88.57	Parallel Bars	0.78	0.88	0.82	87.5
Baseball Pitch	0.92	0.90	0.91	90.38	Pizza Tossing	0.84	0.79	0.81	78.79
Basketball	0.92	1.00	0.96	100.0	Playing Cello	0.91	0.93	0.92	93.48
Basketball Dunk	0.80	0.85	0.82	84.85	Playing Daf	1.00	0.94	0.97	94.12
Bench Press	0.90	1.00	0.95	100.0	Playing Dhol	1.00	1.00	1.00	100.0
Biking	1.00	1.00	1.00	100.0	Playing Flute	0.95	0.95	0.95	94.59
Billiards	1.00	1.00	1.00	100.0	Playing Guitar	1.00	0.98	0.99	97.96
Blow DryHair	1.00	0.75	0.86	75.0	Playing Piano	0.96	0.93	0.94	92.59
Blowing Candles	1.00	0.89	0.94	88.89	Playing Sitar	1.00	0.93	0.96	92.86
Body WeightSquats	0.69	0.62	0.66	62.5	Playing Tabla	1.00	0.97	0.98	96.55
Bowling	0.98	1.00	0.99	100.0	Playing Violin	0.95	0.97	0.96	97.44
Boxing PunchingBag	1.00	0.93	0.96	92.68	Pole Vault	0.85	1.00	0.92	100.0
Boxing SpeedBag	0.96	0.92	0.94	92.0	Pommel Horse	1.00	0.93	0.96	92.86
Breast Stroke	0.83	0.86	0.84	86.36	Pull Ups	0.94	0.77	0.85	77.27
Brushing Teeth	0.71	0.69	0.70	68.57	Punch	0.94	0.96	0.95	96.15
Clean And Jerk	0.96	0.96	0.96	96.3	Push Ups	0.96	0.93	0.95	92.86
Cliff Diving	0.95	1.00	0.97	100.0	Rafting	0.88	0.92	0.90	91.67
Cricket Bowling	0.87	0.89	0.88	89.13	Rock Climbing Indoor	1.00	0.73	0.84	72.73
Cricket Shot	0.93	0.88	0.91	87.76	Rope Climbing	0.90	0.85	0.88	84.85
CuttingIn Kitchen	1.00	0.75	0.86	75.0	Rowing	0.83	1.00	0.90	100.0
Diving	0.96	0.96	0.96	95.56	Salsa Spin	0.83	0.75	0.79	75.0
Drumming	0.91	1.00	0.95	100.0	Shaving Beard	0.76	0.74	0.75	74.42
Fencing	0.95	0.95	0.95	94.87	Shotput	0.77	0.73	0.75	72.97
FieldHockey Penalty	0.93	0.79	0.86	79.41	Skate Boarding	0.91	0.88	0.89	87.5
Floor Gymnastics	0.95	0.95	0.95	94.59	Skiing	1.00	1.00	1.00	100.0
Frisbee Catch	0.97	0.85	0.91	85.37	Skijet	1.00	0.89	0.94	89.29
Front Crawl	0.90	0.88	0.89	88.1	Sky Diving	1.00	1.00	1.00	100.0
Golf Swing	0.85	0.85	0.85	85.19	Soccer Juggling	0.71	0.69	0.70	69.44
Haircut	0.82	0.93	0.87	93.33	Soccer Penalty	0.78	1.00	0.88	100.0
Hammering	0.89	0.95	0.92	95.45	Still Rings	0.96	0.86	0.91	85.71
Hammer Throw	0.87	0.87	0.87	86.67	Sumo Wrestling	0.92	1.00	0.96	100.0
Handstand Pushups	0.91	1.00	0.96	100.0	Surfing	0.96	1.00	0.98	100.0
Handstand Walking	0.88	0.81	0.84	80.77	Swing	0.83	0.97	0.89	97.14
Head Massage	0.83	0.95	0.89	94.59	Table TennisShot	0.93	1.00	0.97	100.0
High Jump	0.82	0.53	0.64	52.94	Tai Chi	0.90	0.81	0.85	81.4
Horse Race	0.97	0.94	0.95	93.94	Tennis Swing	0.84	0.91	0.87	91.11
Horse Riding	0.93	0.98	0.95	98.08	Throw Discus	0.93	0.69	0.79	69.23
Hula Hoop	0.97	0.78	0.87	78.38	Trampoline Jumping	0.92	0.94	0.93	93.75
Ice Dancing	0.98	1.00	0.99	100.0	Typing	0.98	1.00	0.99	100.0
Javelin Throw	0.71	0.82	0.76	81.82	Uneven Bars	0.84	0.90	0.87	90.0
Juggling Balls	0.86	1.00	0.93	100.0	Volleyball Spiking	0.88	0.95	0.91	94.74
Jumping Jack	0.94	0.89	0.92	89.19	Walking With Dog	0.86	0.97	0.91	96.97
Jump Rope	0.00	0.00	0.00	0.0	Wall Pushups	0.90	0.80	0.85	80.0
Kayaking	0.94	0.83	0.88	82.86	Writing On Board	0.96	0.93	0.95	93.48
Knitting	0.98	1.00	0.99	100.0	YoYo	1.00	0.89	0.94	89.19
Long Jump	0.87	0.89	0.88	89.19					

Tabela 10 – Resultados GCN - Limiar = 0.25

GCN - Limiar = 0.25									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.91	0.90	0.91	89.58	Lunges	0.78	0.95	0.86	95.45
Apply Lipstick	0.91	0.94	0.92	93.75	Military Parade	0.97	0.97	0.97	97.37
Archery	0.80	0.86	0.83	86.05	Mixing	1.00	0.90	0.95	89.74
Baby Crawling	0.90	0.95	0.93	95.0	Mopping Floor	0.96	0.81	0.88	80.65
Balance Beam	0.97	0.76	0.85	75.68	Nunchucks	0.95	0.86	0.90	86.36
Band Marching	0.97	0.97	0.97	97.37	Parallel Bars	0.88	1.00	0.94	100.0
Baseball Pitch	0.94	0.94	0.94	93.88	Pizza Tossing	0.75	0.91	0.82	91.3
Basketball	0.95	0.98	0.96	97.56	Playing Cello	0.97	0.95	0.96	95.12
Basketball Dunk	0.88	0.92	0.90	92.31	Playing Daf	1.00	0.96	0.98	96.0
Bench Press	1.00	1.00	1.00	100.0	Playing Dhol	1.00	0.98	0.99	97.56
Biking	1.00	0.94	0.97	93.55	Playing Flute	1.00	1.00	1.00	100.0
Billiards	1.00	1.00	1.00	100.0	Playing Guitar	0.96	1.00	0.98	100.0
Blow DryHair	0.93	0.85	0.89	84.85	Playing Piano	0.94	0.97	0.96	97.06
Blowing Candles	0.97	0.97	0.97	96.67	Playing Sitar	1.00	0.94	0.97	94.44
Body WeightSquats	0.94	0.73	0.82	73.17	Playing Tabla	0.92	1.00	0.96	100.0
Bowling	0.98	0.93	0.96	93.48	Playing Violin	0.93	0.93	0.93	93.33
Boxing PunchingBag	0.88	0.94	0.91	93.88	Pole Vault	0.88	1.00	0.94	100.0
Boxing SpeedBag	0.94	0.97	0.95	96.88	Pommel Horse	1.00	0.92	0.96	92.31
Breast Stroke	0.95	0.83	0.89	83.33	Pull Ups	0.97	0.90	0.93	90.32
Brushing Teeth	0.77	0.79	0.78	79.41	Punch	1.00	1.00	1.00	100.0
Clean And Jerk	1.00	0.97	0.99	97.14	Push Ups	0.97	0.97	0.97	96.67
Cliff Diving	0.95	0.95	0.95	94.74	Rafting	0.97	0.81	0.88	80.56
Cricket Bowling	0.91	0.86	0.89	86.49	Rock Climbing Indoor	1.00	0.83	0.91	83.33
Cricket Shot	0.89	0.94	0.92	94.29	Rope Climbing	0.82	0.90	0.86	90.24
CuttingIn Kitchen	0.94	1.00	0.97	100.0	Rowing	1.00	0.89	0.94	89.47
Diving	1.00	0.98	0.99	98.04	Salsa Spin	0.76	0.94	0.84	93.94
Drumming	1.00	1.00	1.00	100.0	Shaving Beard	0.85	0.80	0.82	79.59
Fencing	0.88	0.97	0.92	96.77	Shotput	0.83	0.78	0.81	78.38
FieldHockey Penalty	0.95	0.86	0.90	85.71	Skate Boarding	0.90	1.00	0.95	100.0
Floor Gymnastics	0.92	0.92	0.92	91.89	Skiing	1.00	1.00	1.00	100.0
Frisbee Catch	0.91	1.00	0.95	100.0	Skijet	1.00	0.96	0.98	96.43
Front Crawl	0.79	1.00	0.88	100.0	Sky Diving	1.00	1.00	1.00	100.0
Golf Swing	0.77	0.86	0.81	86.05	Soccer Juggling	0.83	0.68	0.75	67.86
Haircut	0.87	0.89	0.88	89.19	Soccer Penalty	0.93	0.97	0.95	97.37
Hammering	0.91	0.98	0.95	97.73	Still Rings	0.96	0.93	0.95	92.86
Hammer Throw	0.88	0.86	0.87	86.36	Sumo Wrestling	0.94	0.97	0.96	96.97
Handstand Pushups	0.97	0.92	0.94	91.67	Surfing	0.97	1.00	0.99	100.0
Handstand Walking	0.87	0.69	0.77	68.97	Swing	1.00	0.97	0.99	97.37
Head Massage	0.90	1.00	0.95	100.0	Table TennisShot	0.95	1.00	0.97	100.0
High Jump	0.97	0.87	0.92	86.67	Tai Chi	0.97	0.88	0.92	87.8
Horse Race	0.97	0.94	0.95	93.94	Tennis Swing	0.90	0.91	0.91	91.49
Horse Riding	0.96	1.00	0.98	100.0	Throw Discus	0.81	0.83	0.82	83.33
Hula Hoop	0.82	0.82	0.82	82.5	Trampoline Jumping	0.95	1.00	0.97	100.0
Ice Dancing	1.00	0.98	0.99	97.78	Typing	0.98	1.00	0.99	100.0
Javelin Throw	0.83	0.88	0.86	88.24	Uneven Bars	0.97	0.97	0.97	96.88
Juggling Balls	0.91	0.91	0.91	90.91	Volleyball Spiking	0.92	0.92	0.92	91.67
Jumping Jack	0.92	0.73	0.81	73.33	Walking With Dog	0.96	0.87	0.92	87.1
Jump Rope	0.00	0.00	0.00	0.0	Wall Pushups	0.97	0.93	0.95	93.33
Kayaking	0.82	0.93	0.87	93.1	Writing On Board	0.98	0.98	0.98	97.87
Knitting	1.00	1.00	1.00	100.0	YoYo	0.98	0.98	0.98	97.67
Long Jump	0.85	0.94	0.89	94.44					

Tabela 11 – Resultados GCN - Limiar = 0.50

GCN - Limiar = 0.50									
Classe	Precisão	Revocação	F-score	Acc./Classe	Classe	Precisão	Revocação	F-score	Acc./Classe
Apply EyeMakeup	0.83	0.90	0.86	89.74	Lunges	0.82	0.64	0.72	64.29
Apply Lipstick	0.94	0.85	0.89	85.0	Military Parade	0.97	0.95	0.96	95.0
Archery	0.94	0.83	0.88	83.33	Mixing	1.00	1.00	1.00	100.0
Baby Crawling	0.95	0.98	0.97	97.67	Mopping Floor	1.00	0.81	0.89	80.77
Balance Beam	0.93	0.85	0.89	84.85	Nunchucks	0.95	0.61	0.75	61.29
Band Marching	0.94	0.94	0.94	94.44	Parallel Bars	0.92	0.95	0.93	94.59
Baseball Pitch	0.98	1.00	0.99	100.0	Pizza Tossing	0.74	0.97	0.84	96.55
Basketball	0.97	0.97	0.97	97.3	Playing Cello	1.00	0.98	0.99	97.67
Basketball Dunk	0.69	0.94	0.80	94.29	Playing Daf	0.96	1.00	0.98	100.0
Bench Press	0.93	1.00	0.96	100.0	Playing Dhol	1.00	1.00	1.00	100.0
Biking	0.95	0.98	0.96	97.62	Playing Flute	1.00	0.97	0.98	96.55
Billiards	1.00	1.00	1.00	100.0	Playing Guitar	1.00	0.96	0.98	96.23
Blow DryHair	0.90	0.90	0.90	90.0	Playing Piano	0.98	1.00	0.99	100.0
Blowing Candles	0.94	0.97	0.96	96.97	Playing Sitar	1.00	1.00	1.00	100.0
Body WeightSquats	1.00	0.79	0.88	78.57	Playing Tabla	0.94	0.97	0.95	96.67
Bowling	1.00	0.96	0.98	96.15	Playing Violin	1.00	1.00	1.00	100.0
Boxing PunchingBag	0.93	0.97	0.95	97.5	Pole Vault	0.84	0.97	0.90	97.44
Boxing SpeedBag	0.82	0.97	0.89	96.97	Pommel Horse	1.00	0.67	0.80	66.67
Breast Stroke	0.93	0.84	0.88	83.87	Pull Ups	0.93	0.87	0.90	87.1
Brushing Teeth	0.86	0.83	0.85	83.33	Punch	0.98	0.93	0.95	92.98
Clean And Jerk	1.00	0.93	0.97	93.33	Push Ups	0.94	0.97	0.95	96.77
Cliff Diving	0.80	1.00	0.89	100.0	Rafting	0.96	1.00	0.98	100.0
Cricket Bowling	0.93	0.70	0.80	70.0	Rock Climbing Indoor	1.00	0.80	0.89	80.0
Cricket Shot	0.84	0.94	0.89	94.12	Rope Climbing	0.71	0.88	0.79	88.24
CuttingIn Kitchen	1.00	1.00	1.00	100.0	Rowing	1.00	0.90	0.95	90.0
Diving	0.96	1.00	0.98	100.0	Salsa Spin	0.86	0.94	0.90	94.12
Drumming	0.98	1.00	0.99	100.0	Shaving Beard	0.91	0.85	0.88	85.29
Fencing	0.95	0.88	0.91	87.5	Shotput	0.84	0.80	0.82	80.0
FieldHockey Penalty	0.89	0.82	0.85	82.05	Skate Boarding	0.82	0.97	0.89	96.55
Floor Gymnastics	0.94	0.88	0.91	87.88	Skiing	1.00	0.92	0.96	92.11
Frisbee Catch	0.84	0.94	0.89	94.12	Skijet	1.00	0.88	0.94	88.0
Front Crawl	0.89	0.94	0.91	94.12	Sky Diving	0.92	0.92	0.92	92.0
Golf Swing	0.81	0.92	0.86	91.89	Soccer Juggling	0.70	0.76	0.73	76.19
Haircut	0.93	0.93	0.93	92.68	Soccer Penalty	0.98	0.94	0.96	94.12
Hammering	0.96	0.96	0.96	95.56	Still Rings	0.94	0.97	0.96	96.97
Hammer Throw	0.89	0.87	0.88	86.96	Sumo Wrestling	1.00	1.00	1.00	100.0
Handstand Pushups	0.95	1.00	0.97	100.0	Surfing	0.89	1.00	0.94	100.0
Handstand Walking	0.59	0.87	0.70	87.1	Swing	0.91	0.97	0.94	97.5
Head Massage	0.92	1.00	0.96	100.0	Table TennisShot	1.00	0.95	0.98	95.24
High Jump	0.85	0.79	0.81	78.57	Tai Chi	0.89	0.94	0.92	94.29
Horse Race	1.00	0.94	0.97	94.29	Tennis Swing	0.93	0.91	0.92	91.3
Horse Riding	0.97	1.00	0.99	100.0	Throw Discus	0.72	0.82	0.77	81.58
Hula Hoop	0.95	0.66	0.78	65.62	Trampoline Jumping	0.86	0.95	0.90	94.74
Ice Dancing	1.00	1.00	1.00	100.0	Typing	1.00	1.00	1.00	100.0
Javelin Throw	0.96	0.65	0.77	64.71	Uneven Bars	0.89	1.00	0.94	100.0
Juggling Balls	0.95	0.95	0.95	94.74	Volleyball Spiking	0.80	0.91	0.85	91.43
Jumping Jack	0.97	0.90	0.93	89.74	Walking With Dog	0.89	0.78	0.83	78.12
Jump Rope	0.00	0.00	0.00	0.0	Wall Pushups	0.92	0.92	0.92	91.67
Kayaking	0.98	0.98	0.98	97.67	Writing On Board	0.98	0.98	0.98	97.56
Knitting	1.00	1.00	1.00	100.0	YoYo	1.00	0.90	0.95	89.58
Long Jump	0.93	0.93	0.93	93.18					

Tabela 12 – Sumarização das métricas globais

	Precisão		Revocação		F-score		Acurácia
Modelo	Med. Ponderada	Med. Macro	Med. Ponderada	Med. Macro	Med. Ponderada	Med Macro	
Conv 3D	0.28	0.28	0.29	0.28	0.28	0.27	29%
C3D	0.34	0.33	0.33	0.31	0.31	0.30	33%
LRCN	0.21	0.20	0.24	0.23	0.21	0.20	24%
GCN - Sequencial	0.61	0.60	0.62	0.61	0.59	0.62	62%
GCN - Limiar 0.15	0.90	0.89	0.90	0.88	0.90	0.89	90%
GCN - Limiar 0.25	0.92	0.92	0.92	0.92	0.92	0.92	92%
GCN - Limiar 0.50	0.92	0.92	0.92	0.92	0.92	0.92	92%

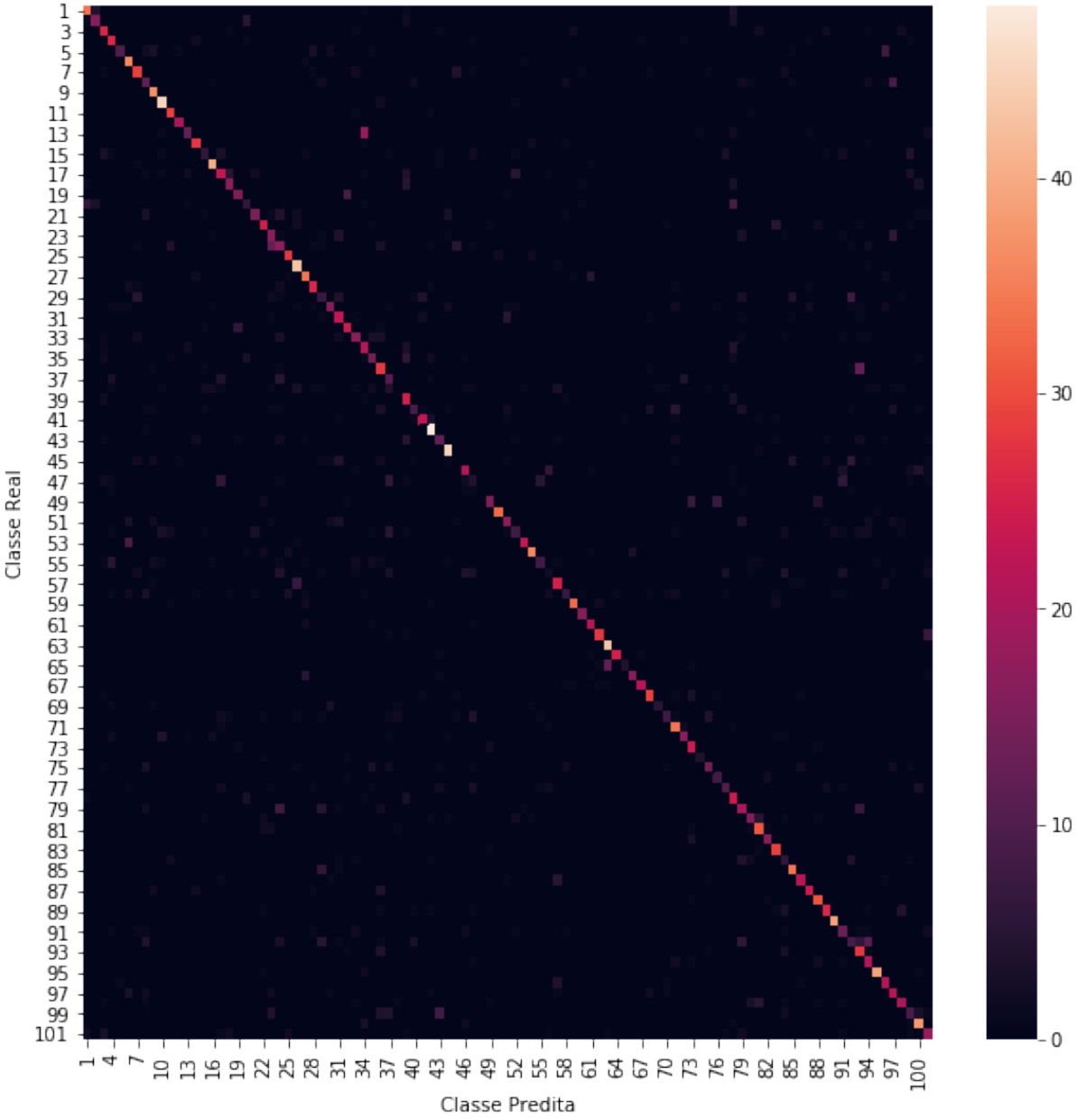

Figura 16 – Mapa de calor da matriz de confusão para modelo GCN - sequencial

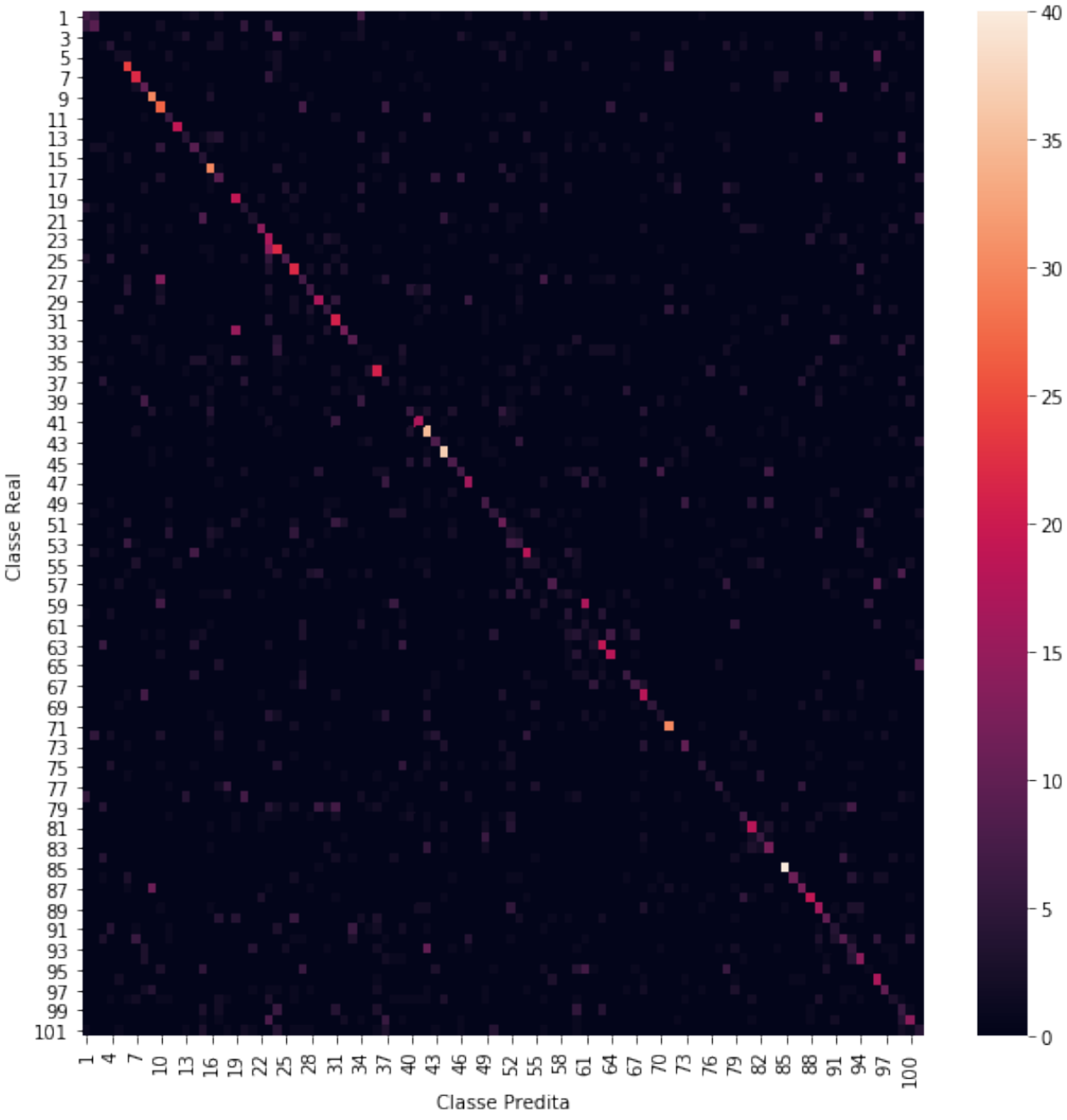

Figura 17 – Mapa de calor da matriz de confusão para modelo Convolucional 3D

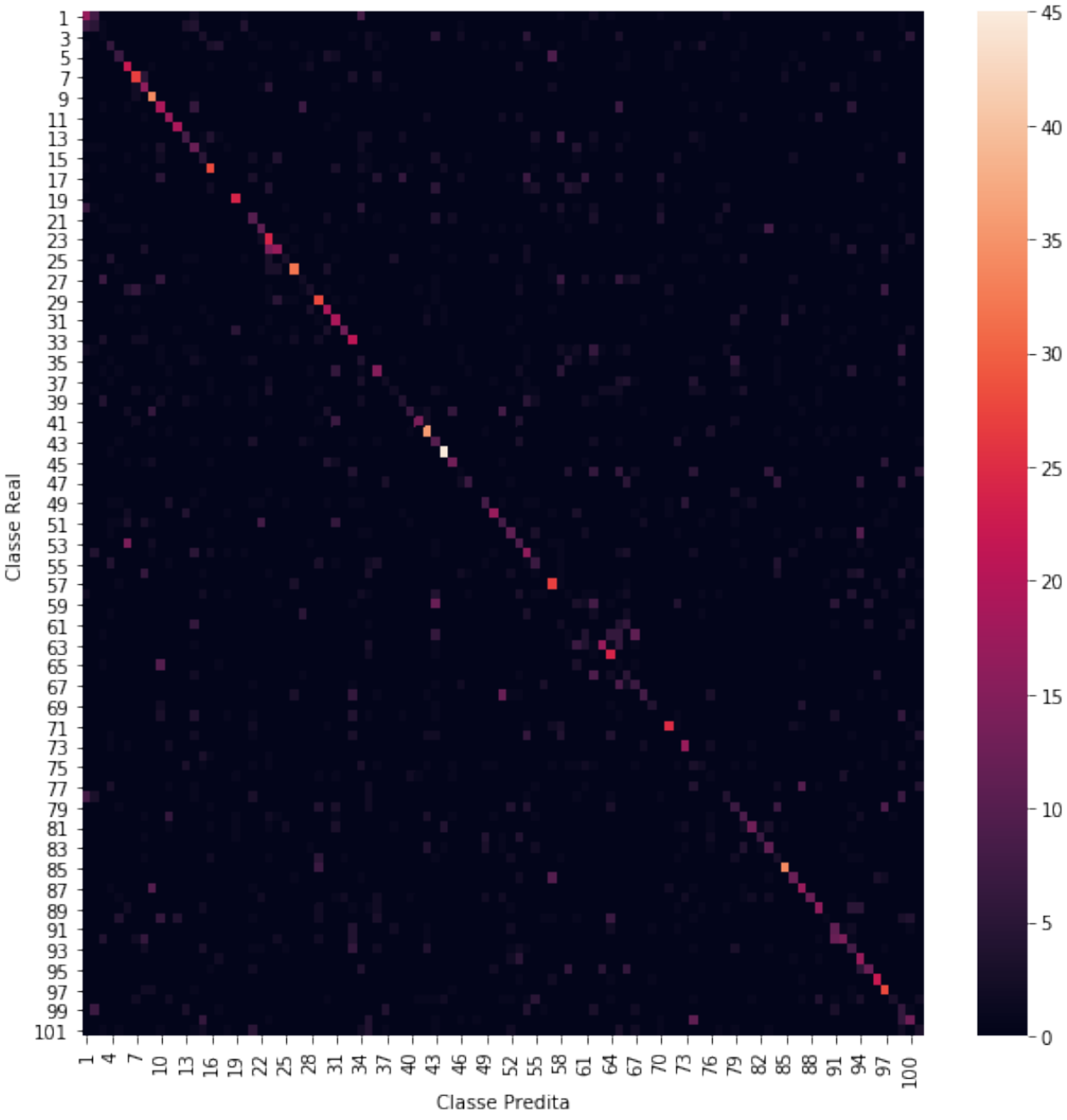

Figura 18 – Mapa de calor da matriz de confusão para modelo C3D

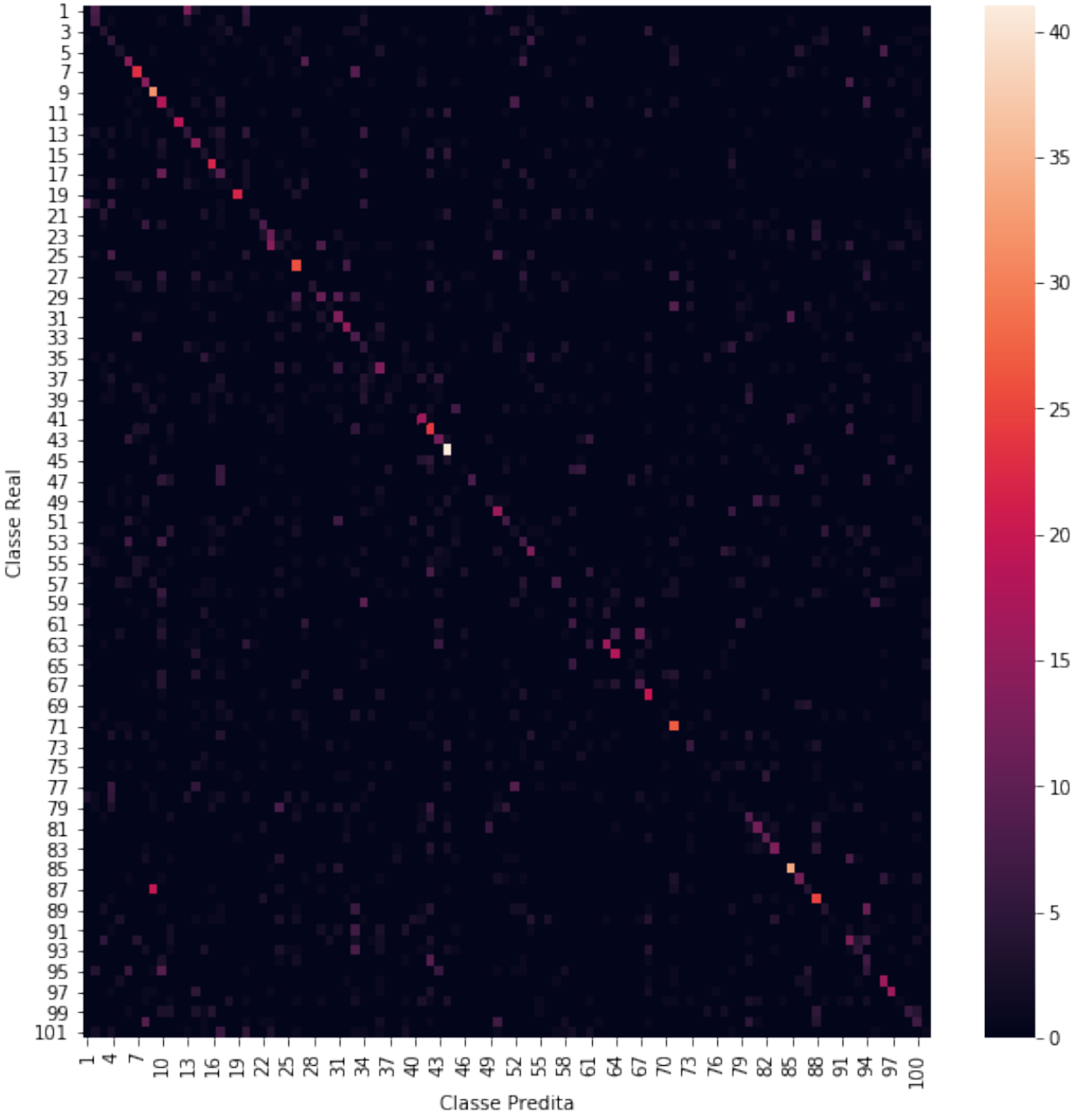

Figura 19 – Mapa de calor da matriz de confusão para modelo LRCN

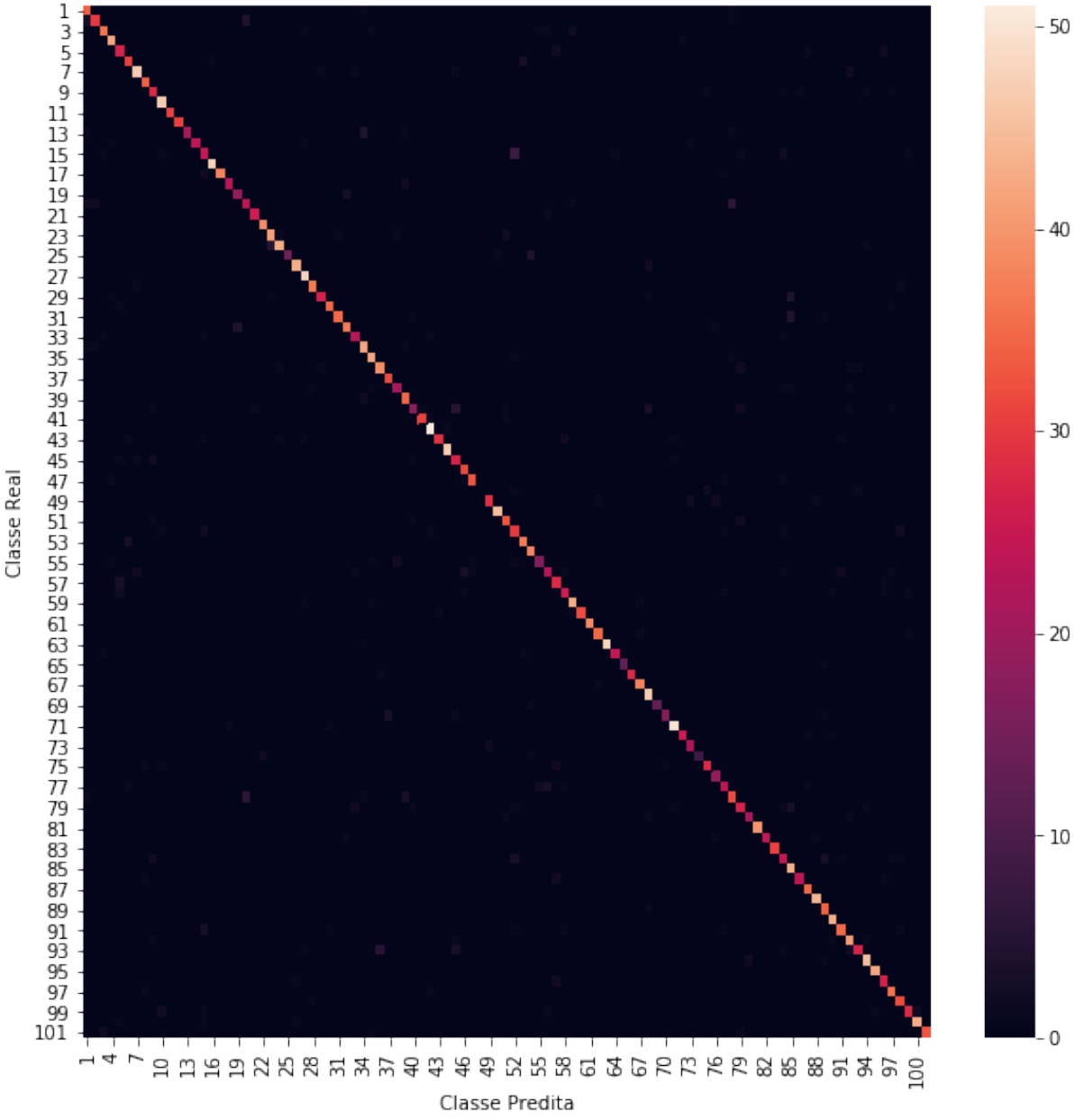

Figura 20 – Mapa de calor da matriz de confusão para modelo GCN - Limiar 0.15

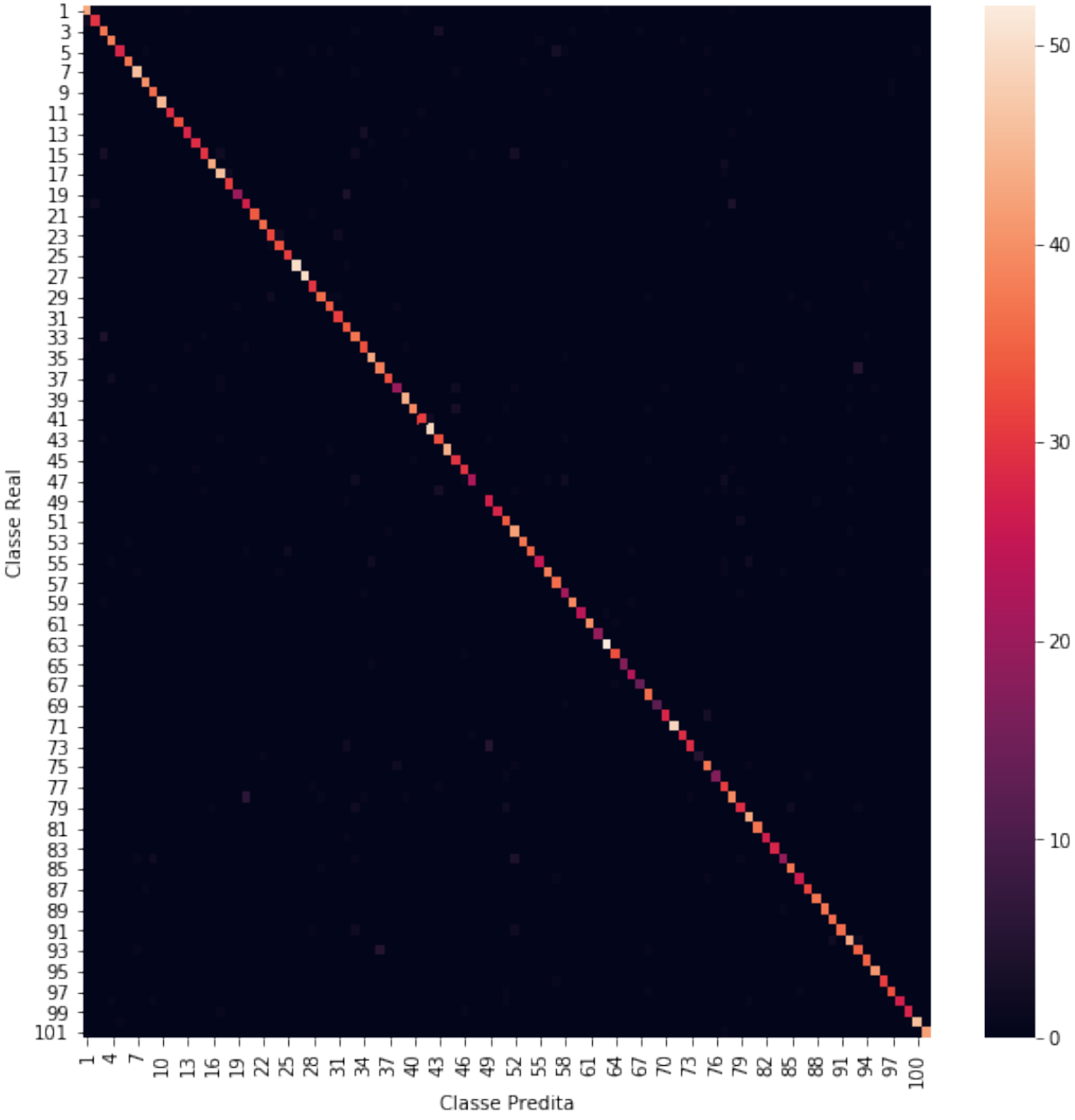

Figura 21 – Mapa de calor da matriz de confusão para modelo GCN - Limiar 0.25

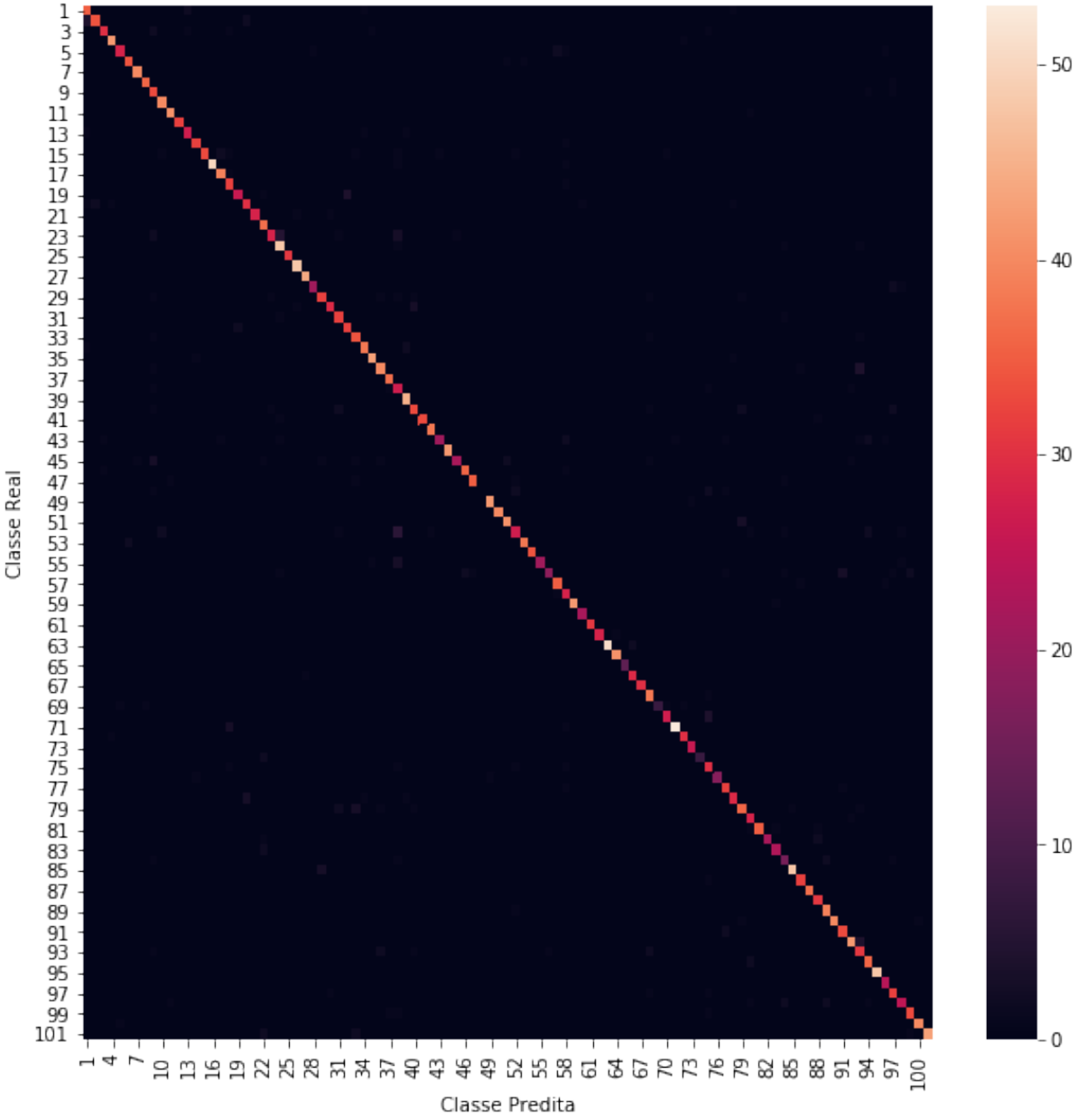

Figura 22 – Mapa de calor da matriz de confusão para modelo GCN - Limiar 0.50

4 CONCLUSÕES

O reconhecimento de ações em vídeo demonstra-se uma atividade complexa e de alto custo computacional. O mesmo demanda técnicas muitas vezes computacionalmente onerosas pois pode gerar modelos volumosos. Assim, é importante conseguir um balanceamento entre eficiência e eficácia.

O método proposto no presente trabalho mostrou uma boa eficácia na classificação de ações em vídeos com uso de GCNs. Para tanto, foram geradas conexões de grafos de forma simples e com baixo custo computacional, em vista de ser um modelo que é carregado totalmente em memória. Quando comparado a modelos baseados em filtros convolucionais tri-dimensionais como C3D e variações, o método proposto provê uma maior elasticidade de representação da base amostral e menor custo de execução por épocas de treinamento.

Relacionado especificamente aos resultados, uma quantidade considerável de classes apresentaram acima de 80% em precisão e revocação. Além disso, não houveram muitas classes com métricas muito abaixo do esperado, mantendo uma precisão de 60% no mínimo.

Apesar dos fatores citados acima, alguns impedimentos também puderam ser constatados. Um deles diz respeito ao tempo para gerar os grafos, visto que é um processo oneroso. No entanto, trata-se de um processo *offline* sendo que o mesmo é gerado uma única vez. Claro, caso haja aumento incremental da base o grafo deve ser gerado novamente, mas esse fator afeta qualquer outro método de classificação.

Assim, de maneira geral, o método proposto mostrou-se promissor ao ser comparado com técnicas concorrentes. Além disso, o mesmo abre muitas possibilidades de modificações, melhorias e análises futuras.

4.1 TRABALHOS FUTUROS

Como trabalhos futuros foram elencados os seguintes pontos:

1. Carregamento da arquitetura em lotes, sem a necessidade de carregamento total em memória;

2. Proposta de conexões serem ponderadas automaticamente em tempo de execução durante o processo de treinamento;

3. Proposta de diferentes estratégias de conexão dos nós de acordo com o contexto do problema;
4. Proposta de uma abordagem para definição adequada de limiar de poda de conexões;
5. Aplicar o método proposto em outros datasets de vídeos tanto para reconhecimento de ações, como envolvendo outros contextos em vídeos;
6. Considerar abordagens híbridas de GCN com modelos CNN acoplados diretamente.

4.2 PUBLICAÇÕES

Esse trabalho, até o atual momento da escrita do presente texto, gerou a submissão a periódico (Qualis B1) do seguinte artigo:

- Costa, F. F., Saito, P. T. M., Bugatti, P. H. *Video Action Classification through Graph Convolutional Networks*. Machine Vision and Applications. pp. 1-11. 2020 (*under review*).

REFERÊNCIAS

BACCOUCHE, M.; MAMALET, F.; WOLF, C.; GARCIA, C.; BASKURT, A. Action classification in soccer videos with long short-term memory recurrent neural networks. *In*: DIAMANTARAS, Konstantinos; DUCH, Wlodek; ILIADIS, Lazaros S. (Ed.). **Artificial Neural Networks – ICANN 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 154–159. ISBN 978-3-642-15822-3.

BACCOUCHE, M.; MAMALET, F.; WOLF, C.; GARCIA, C.; BASKURT, A. Sequential deep learning for human action recognition. *In*: SALAH, Albert Ali; LEPRI, Bruno (Ed.). **Human Behavior Understanding**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 29–39. ISBN 978-3-642-25446-8.

BRUNA, J.; ZAREMBA, W.; SZLAM, A.; LECUN, Y. Spectral networks and locally connected networks on graphs. *In*: **International Conference on Learning Representations (ICLR2014), CBLS, April 2014**. *[S.l.: s.n.]*, 2014.

DENG, L.; YU, D. **Deep Learning: Methods and Applications**. Hanover, MA, USA: Now Publishers Inc., 2014. ISBN 1601988141, 9781601988140.

DOLLAR, P.; RABAUD, V.; COTTRELL, G.; BELONGIE, S. Behavior recognition via sparse spatio-temporal features. *In*: **Proceedings of the 14th International Conference on Computer Communications and Networks**. Washington, DC, USA: IEEE Computer Society, 2005. (ICCCN '05), p. 65–72. ISBN 0-7803-9424-0. Disponível em: http://dl.acm.org/citation.cfm?id=1259587.1259830.

DONAHUE, J.; HANDRIKS, L. A.; ROHRBACH, M.; VENUGOPALAN, S.; GUADARRAMA, S.; SAENKO, K.; DARRELL, T. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. **arXiv e-prints**, p. arXiv:1411.4389, nov. 2014.

DUVENAUD, D. K.; MACLAURIN, D.; IPARRAGUIRRE, J.; BOMBARELL, R.; HIRZEL, T.; ASPURU-GUZIK, A.; ADAMS, R. P. Convolutional networks on graphs for learning molecular fingerprints. *In*: CORTES, C.; LAWRENCE, N. D.; LEE, D. D.; SUGIYAMA, M.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 28**. Curran Associates, Inc., 2015. p. 2224–2232. Disponível em: http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints.pdf.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological cybernetics**, v. 36, p. 193–202, 02 1980.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. Disponível em: http://www.deeplearningbook.org.

HAYKIN, S.S. **Redes Neurais - 2ed.** Bookman, 2001. ISBN 9788573077186. Disponível em: https://books.google.com.br/books?id=lBp0X5qfyjUC.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. **arXiv e-prints**, p. arXiv:1512.03385, dez. 2015.

HENAFF, M.; BRUNA, J.; LECUN, Y. Deep convolutional networks on graph-structured data. **CoRR**, abs/1506.05163, 2015. Disponível em: http://arxiv.org/abs/1506.05163.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. Disponível em: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

JAIN, A.; ZAMIR, A. R.; SAVARESE, S.; SAXENA, A. Structural-rnn: Deep learning on spatio-temporal graphs. **CoRR**, abs/1511.05298, 2015. Disponível em: http://arxiv.org/abs/1511.05298.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: A review. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, set. 1999. ISSN 0360-0300. Disponível em: http://doi.acm.org/10.1145/331499.331504.

JAIN, M.; JÉGOU, H.; BOUTHEMY, P. Better exploiting motion for better action recognition. *In*: **2013 IEEE Conference on Computer Vision and Pattern Recognition**. *[S.l.: s.n.]*, 2013. p. 2555–2562. ISSN 1063-6919.

JI, S.; XU, W.; YANG, M.; YU, K. 3d convolutional neural networks for human action recognition. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 35, n. 1, p. 221–231, jan. 2013. ISSN 0162-8828. Disponível em: http://dx.doi.org/10.1109/TPAMI.2012.59.

KARPATHY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; FEI-FEI, L. Large-scale video classification with convolutional neural networks. *In*: **Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition**. Washington, DC, USA: IEEE Computer Society, 2014. (CVPR '14), p. 1725–1732. ISBN 978-1-4799-5118-5. Disponível em: https://doi.org/10.1109/CVPR.2014.223.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2014.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *In*: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.;

WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 25**. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LAPTEV, I. On space-time interest points. **International Journal of Computer Vision**, v. 64, n. 2, p. 107–123, Sep 2005. ISSN 1573-1405. Disponível em: https://doi.org/10.1007/s11263-005-1838-7.

LAPTEV, I.; MARSZALEK, M.; SCHMID, C.; ROZENFELD, B. Learning realistic human actions from movies. *In*: **2008 IEEE Conference on Computer Vision and Pattern Recognition**. *[S.l.: s.n.]*, 2008. p. 1–8. ISSN 1063-6919.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *In*: **Proceedings of the IEEE**. *[s.n.]*, 1998. v. 86, n. 11, p. 2278–2324. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665.

LI, Y.; T., DANIEL; B., MARC; Z, RICHARD S. Gated graph sequence neural networks. **CoRR**, abs/1511.05493, 2016.

MARR, D. **Vision: A Computational Investigation into the Human Representation and Processing of Visual Information**. New York, NY, USA: Henry Holt and Co., Inc., 1982. ISBN 0716715678.

MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 127–147, 1943.

ROSENBLATT, F. .the perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, p. 65–386, 1958.

RUSSAKOVSKY, O.; D., JIA; S., HAO; K., JONATHAN; S., SANJEV; M., SEAN; H., ZHIHENG; K., AANDREJ; K., ADITYA; B., MICHAEL; B, ALEXANDER C.; F., LI. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

SIMONYAN, K.; ZISSERMAN, A. Two-stream convolutional networks for action recognition in videos. *In*: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N. D.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 27**. Curran Associates, Inc., 2014. p. 568–576. Disponível em: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf.

SMITH, S.W. **The Scientist and Engineer's Guide to Digital Signal Processing**. California Technical Pub., 1997. ISBN 9780966017632. Disponível em: https://books.google.com.br/books?id=rp2VQgAACAAJ.

SOOMRO, K.; ZAMIR, A. R.; SHAH, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. **CoRR**, abs/1212.0402, 2012. Disponível em: http://dblp.uni-trier.de/db/journals/corr/corr1212.html#abs-1212-0402.

TRAN, D.; BOURDEV, L.; FERGUS, R.; TORRESANI, L.; PALURI, M. Learning Spatiotemporal Features with 3D Convolutional Networks. **arXiv e-prints**, p. arXiv:1412.0767, dez. 2014.

WANG, H.; KLÄSER, A.; SCHMID, C.; LIU, C. Action recognition by dense trajectories. *In*: **CVPR 2011**. *[S.l.: s.n.]*, 2011. p. 3169–3176. ISSN 1063-6919.

WANG, H.; ULLAH, M. M.; KLASER, A.; LAPTEV, I.; SCHMID C., year = 2009 pages = 124.1-124.11 booktitle = Proc. BMVC isbn = 1-901725-39-1 note = doi:10.5244/C.23.124. Evaluation of local spatio-temporal features for action recognition. *In*: . *[S.l.: s.n.]*.

WILLEMS, G.; TUYTELAARS, T.; GOOL, L.. V. An efficient dense and scale-invariant spatio-temporal interest point detector. *In*: FORSYTH, David; TORR, Philip; ZISSERMAN, Andrew (Ed.). **Computer Vision – ECCV 2008**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 650–663. ISBN 978-3-540-88688-4.

YUE-HEI NG, J.; HAUS.KNECHT, M.; VIJAYANARASIMHAN, S.; VINYALS., O.; MONGA, R.; TODERICI, G.