UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Detecção de Mentiras a Partir de Vídeos com Deep Learning

Trabalho de Conclusão de Curso

Mateus Karvat Camara

Cascavel–PR

2021

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Mateus Karvat Camara

Detecção de Mentiras a Partir de Vídeos com Deep Learning

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador(a): Adriana Postal

Cascavel–PR

2021

MATEUS KARVAT CAMARA

DETECÇÃO DE MENTIRAS A PARTIR DE VÍDEOS COM DEEP LEARNING

Monografia apresentada como requisito parcial para obtençãoo do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Adriana Postal (Orientadora)
Colegiado de Ciência da Computação, UNIOESTE

Prof. Josué Pereira de Castro
Colegiado de Ciência da Computação, UNIOESTE

Prof. Gustavo Henrique Paetzold
Engenharia da Computação, UTFPR - Toledo

Cascavel, 25 de Julho de 2022

Agradecimentos

Aos meus pais e à minha irmã, pelo apoio prestado de todas as formas possíveis durante todos esses anos, sem o qual eu jamais teria chegado até aqui nem seria a pessoa que sou hoje. O mérito desse trabalho é todo seu.

Aos meus amigos, os quais sou extremamente afortunado de ter em número tão grande a ponto de ser inviável listá-los aqui. Vocês fazem a vida valer a pena.

Aos mestres, por todas as lições oferecidas, principalmente as que vão além da formação técnica e abarcam a formação humana. Em especial, aos professores Marcio Seiji Oyamada, Adriana Postal e Tomás Henrique Maul pela oportunidade de enriquecer enormemente minha formação.

Ao MEC-SESU, pelo financiamento de projetos de iniciação científica a partir do Programa de Educação Tutorial (PET) que levaram ao desenvolvimento desse trabalho.

Tired of lying in the sunshine, staying home to watch the rain
You are young and life is long, and there is time to kill today
And then one day you find ten years have got behind you
No one told you when to run, you missed the starting gun
(Pink Floyd)

Resumo

A detecção de mentiras é uma tarefa em que seres humanos apresentam notável dificuldade, alcançando acurácia de apenas 54% de acordo com a literatura acerca do tema. Apesar disso, tal tarefa adquire grande relevância em contextos como tribunais, entrevistas e investigações criminais, nos quais o impacto da má classificação de um discurso como mentiroso ou verdadeiro pode ser catastrófico. Nesse sentido, mentiras e sua identificação atraem a atenção de pesquisadores há séculos, com a criação de dispositivos destinados a auxiliar em sua detecção e, mais recentemente, o desenvolvimento de sistemas de *Machine Learning* e *Deep Learning* capazes de identificá-las apropriadamente. Tomando por base tais sistemas, o presente trabalho investiga a literatura existente acerca do tema e implementa uma Rede Neural Profunda a partir da arquitetura SlowFast com utilização do *dataset* Real-Life Trial alcançando acurácia de 66,36%. Destaque especial é dado à discussão de questões éticas e limitações da utilização de sistemas de *Machine Learning* e *Deep Learning* que realizam detecção de mentiras, recomendando-se que tais sistemas não sejam utilizados em contextos reais dadas as limitações atualmente existentes para seu desenvolvimento adequado.

Palavras-chave: Detecção de mentiras. *Deep Learning*. Reconhecimento de vídeos. SlowFast. Real-Life Trial Dataset.

Lista de figuras

Lista de quadros

Lista de tabelas

Lista de códigos

Lista de abreviaturas e siglas

3DCNN	*Tridimensional Convolutional Neural Network* (Rede Neural Convolucional Tridimensional)
ANN	*Artificial Neural Network* (Rede Neural Artificial)
AUC	*Area Under Curve* (Área Sob a Curva)
CMC	Comunicação Mediada por Computador
CNN	*Convolutional Neural Network* (Rede Neural Convolucional)
DD	*Deception Detection* (Detecção de Mentiras)
DL	*Deep Learning* (Aprendizagem Profunda)
EEG	Eletroencefalograma
fNIRS	*Functional Near-Infrared Spectroscopy* (Espectroscopia Funcional em Infravermelho Próximo)
LSTM	*Long Short-Term Memory*
ML	*Machine Learning* (Aprendizagem de Máquina)
NL	*Non-Local*
OF	*Optical Flow* (Fluxo Ótico)
RF	*Random Forest*
RLT	*Real-Life Trial*
RNN	*Recurrent Neural Network* (Rede Neural Recorrente)
SGD	*Stochastic Gradient Descent*
SVM	*Support Vector Machine*

Sumário

1

Introdução

Mentiras estão presentes nas sociedades humanas nas mais diversas esferas e contextos, podendo ser observadas desde corriqueiras interações cotidianas até negociações cujos resultados afetam milhões de vidas (VRIJ, 2008). Ainda que em muitas situações o impacto causado por mentiras possa ser pequeno ou mesmo negligenciável, em contextos como investigações criminais e política internacional seus efeitos podem ser catastróficos.

Nesse sentido, a identificação de mentiras é uma tarefa que intriga a humanidade há séculos (KHAN et al., 2021), visto que pode se mostrar extremamente difícil em determinados contextos. Quando uma mentira visa falsificar um fato do qual o ouvinte tem conhecimento, sua identificação torna-se trivial. Em muitos casos, quando uma pessoa que conhecemos bem mente, podemos perceber pelo modo como ela se comunica que está mentindo, devido à convivência com ela. Todavia, como identificar uma mentira contada por um desconhecido que relata um fato do qual não temos conhecimento algum?

A aparente inabilidade humana em identificar mentiras de modo preciso e consistente motivou a criação de máquinas como o polígrafo, as quais são capazes de efetuar tal tarefa de modo supostamente infalível. Todavia, com o tempo, tais máquinas se mostraram passíveis de erro e facilmente ludibriáveis por indivíduos mal intencionados (FIEDLER; SCHMID; STAHL, 2002).

Devido a isso, a ascensão da área de ML (*Machine Learning*, Aprendizagem de Máquina) atraiu a atenção de pesquisadores para o campo de DD (*Deception Detection*, Detecção de Mentiras), buscando resultados confiáveis para tarefa de tamanha relevância. Sistemas foram desenvolvidos a partir de dados de diferentes modalidades, como vídeo, áudio, texto e até mesmo EEG (Eletroencefalograma). Todavia, tais sistemas apresentam limitações que colocam em cheque o aspecto ético de seu desenvolvimento e utilização em contextos reais.

Com base nisso, o presente trabalho tem como principal objetivo implementar uma ANN (*Artificial Neural Network*, Redes Neural Artificial) profunda para DD com base em dados em

formato de vídeo capaz de alcançar acurácia superior à humana.

1.1 Objetivos Específicos

- Breve revisão das técnicas historicamente utilizadas para DD;
- Análise de trabalhos correlatos que realizam DD a partir de ML;
- Seleção de um *dataset* de vídeos contendo humanos mentindo e falando a verdade;
- Investigação acerca das implicações éticas da utilização de um sistema de ML para DD;
- Realização de ajustes na ANN utilizada para otimização dos resultados obtidos;
- Análise dos resultados obtidos pela implementação da ANN.

1.2 Organização

A fim de alcançar os objetivos supracitados, uma revisão conceitual é realizada, estabelecendo a definição de mentira, a acurácia humana em sua identificação, bem como a análise de dispositivos primitivos utilizados para DD. Tal revisão é apresentada no Capítulo 2.

Em seguida, o Capítulo 3 busca analisar trabalhos correlatos que realizam DD a partir de métodos de ML. Primeiramente, métodos baseados em modalidades como áudio, textos e EEG são considerados, tendo seus resultados comparados e as limitações de sua utilização discutidas. Posteriormente, sistemas baseados em vídeo são comparados e analisados, com destaque especial sendo dado para aqueles que utilizam um *dataset* (base de dados) comum. Por fim, as limitações atualmente existentes para sistemas de ML para DD são discutidas, bem como as questões éticas oriundas da construção e utilização de tais modelos.

A fim de selecionar uma arquitetura de DL (*Deep Learning*, Aprendizagem Profunda) capaz de equilibrar baixo custo computacional com alta acurácia, o Capítulo 4 analisa trabalhos de revisão bibliográfica da área de reconhecimento de vídeos, reunindo informações sobre o atual estado da área e comparando arquiteturas consideradas estado da arte nessa tarefa. Após tal comparação, uma arquitetura específica é selecionada e seu funcionamento é descrito.

Com base nas análises realizadas, o Capítulo 5 descreve a metodologia adotada para a implementação da ANN para DD, discutindo as ferramentas utilizadas, os passos realizados para preparar o *dataset* para a etapa de treino, além dos hiperparâmetros de treino avaliados.

Os resultados da implementação da ANN são apresentados, discutidos e analisados no Capítulo 6. Por fim, o Capítulo 7 resume as principais lições obtidas na realização do presente trabalho, indicando potenciais ações a serem adotadas por trabalhos futuros.

2

Detecção de Mentiras

O que caracteriza uma mentira? Quão bem os humanos conseguem identificá-las? E quão confiável é o polígrafo? O presente capítulo discute tais questionamentos, tal que a Seção 2.1 define o conceito de mentiras, discute os tipos de mentiras existentes bem como a frequência com que elas são contadas. A partir disso, a Seção 2.2 discute estudos que buscam determinar a acurácia humana na tarefa de DD. Em seguida, o polígrafo, seu funcionamento e limitações são discutidos na Seção 2.3.

2.1 Mentiras

Uma mentira pode ser definida, de acordo com Vrij (2008), como uma tentativa deliberada e sem prévio aviso, bem-sucedida ou não, de criar em outra pessoa uma crença que o comunicador considera falsa. Tal definição situa mentiras num contexto comunicativo (verbal ou não-verbal) envolvendo um comunicador e um ouvinte, em que a falsidade da informação está intimamente associada à crença do comunicador, de modo que uma mentira pode ocorrer ainda que a informação comunicada seja, em si, verdadeira. Um exemplo dado por Vrij (2008) é de um indivíduo que está escondendo um amigo em sua casa, o qual é suspeito de um crime. Suponha que a polícia vá até a casa do indivíduo e, notando a chegada da polícia, o amigo fuja da casa. Ao dizer à polícia que o amigo não está em sua casa, o indivíduo comunica uma informação verdadeira, mas a qual ele acredita ser falsa e a qual espera que a polícia acredite, portanto caracterizando uma mentira.

Mentiras podem ser classificadas entre baixo impacto (tradução livre do inglês *low-stakes*) ou alto impacto (tradução livre do inglês *high-stakes*) (VRIJ, 2008; NGÔ et al., 2018), sendo aqui o impacto referente às consequências negativas ocasionadas pela descoberta da mentira ou das consequências positivas ocasionadas pela aceitação da mentira (MANN; VRIJ; BULL, 2004). Entre mentiras de alto impacto podemos considerar testemunhos falsos em investigações

policiais, qualificações falsas em entrevistas de emprego ou mesmo falsidades envolvendo infidelidade amorosa. Já as mentiras de baixo impacto são aquelas contadas cotidianamente e cujas consequências têm baixa relevância, sendo chamadas em Língua Inglesa de *white lies*. Dentre tal categoria de mentiras situam-se aquelas contadas com o propósito de provocar uma reação positiva no ouvinte, como afirmações a respeito da aparência de uma pessoa.

Por sua vez, a frequência com que as pessoas mentem foi objeto de estudo de DePaulo et al. (1996), os quais realizaram um experimento com 77 estudantes universitários e 70 membros da sociedade civil que deveriam registrar em um diário as mentiras contadas em seu cotidiano. A conclusão alcançada pelo experimento foi de que as pessoas mentem cerca de 2 vezes por dia. Todavia, Vrij (2008) levanta problemas metodológicos em tal experimento e em outros similares que buscaram atestar a frequência com que mentiras são contadas no cotidiano. Ainda que os estudos não tenham chegado a um número preciso, todos concluíram que, de modo geral, mentiras são frequentemente e cotidianamente contadas.

2.2 Acurácia Humana na Detecção de Mentiras

A identificação de mentiras atrai interesse de pesquisadores e estudiosos há séculos (KHAN et al., 2021) não apenas pelo seu alto impacto na sociedade (GROSS; SHAFFER, 2012), mas também por se tratar de uma tarefa na qual humanos são notavelmente inaptos.

Ekman e O'Sullivan (1991) realizaram um experimento com diferentes grupos compostos por supostos especialistas em DD (membros do serviço secreto, especialistas em polígrafos, investigadores, juízes e psiquiatras) a fim de determinar se tais grupos apresentariam maior sucesso que não-especialistas. Nele, os participantes assistiram a um vídeo em que diferentes pessoas contavam mentiras ou verdades. Deste modo, foi possível comparar o percentual de acerto de membros de tais grupos com o percentual de estudantes universitários, encontrando valores bastante similares que variaram de 52,82% entre os estudantes a 57,61% entre os juízes. Todavia, membros do serviço secreto apresentaram percentual significativamente maior (64,12%) que os demais grupos, o que é justificado pelos autores com a hipótese de que tais indivíduos, devido a seu trabalho de proteção de governantes em multidões, tiveram de refinar suas habilidades de observação e identificação de pessoas potencialmente perigosas, o que os tornou mais aptos a identificar mentiras.

Posteriormente, Mann, Vrij e Bull (2004) realizaram um estudo com 99 policiais, os quais deveriam identificar mentiras a partir de vídeos registrados em tribunais, sendo essas, portanto, mentiras de alto impacto. A acurácia obtida para tais policiais foi de 65%, valor que, comparado aos resultados obtidos por Ekman e O'Sullivan (1991), demonstra uma divergência metodológica e um desafio para se alcançar uma medida precisa da real acurácia humana na identificação de mentiras.

Nesse sentido, Bond e DePaulo (2006) realizaram um metaestudo analisando 206 trabalhos

distintos acerca do tema, os quais totalizaram 24.483 receptores (indivíduos responsáveis por detectar as mentiras), 6.651 mensagens (mentiras ou verdades) e 4.435 comunicadores (indivíduos responsáveis por contar mentiras ou verdades). A partir de tais estudos, os autores chegaram à marca de 54% como o percentual de acerto médio humano para DD, sendo 47% nossa acurácia para classificar mentiras e 61% nossa acurácia para classificar verdades. Tal disparidade reforça a existência, já identificada previamente (MANN; VRIJ; BULL, 2004), de uma tendência humana a avaliar comunicações como verdadeiras, visto que esperamos nos deparar com mais verdades que mentiras em nosso dia-a-dia (VRIJ, 2008). Além disso, os autores afirmam que a real acurácia humana no cotidiano é certamente menor, dada essa expectativa de que a maior parte das comunicações cotidianas sejam verdadeiras e o fato de que num contexto de testagem os participantes do experimento estão geralmente predispostos a identificar mentiras, o que melhora sua taxa de acerto em tal tarefa.

Ademais, Bond e DePaulo (2006) avaliaram, em seu metaestudo, a acurácia humana a partir de diferentes meios de comunicação, apontando que em comunicações exclusivamente auditivas ela é de 53,75%, sendo de 50,35% para comunicações exclusivamente visuais e de 53,98% para comunicações que combinam elementos visuais e auditivos. Eles também compararam o desempenho entre especialistas e não-especialistas, atestando que a diferença absoluta entre tais grupos é insignificante, sendo de cerca de 0,5 pontos percentuais.

Dado que a porcentagem de acerto humana em DD é ligeiramente superior ao total acaso (50%) e dada a grande relevância que mentiras de alto impacto têm na sociedade, afetando investigações criminais, pedidos de crédito, entrevistas de emprego e relacionamentos interpessoais, tem-se um contexto propício para o surgimento de mecanismos capazes de melhorar tal percentual.

2.3 Sistemas Primitivos para Detecção de Mentiras

O interesse por DD data do século VII a.C. (KHAN et al., 2021), com o desenvolvimento de máquinas para detecção automática de mentiras ocorrendo em 1880 na Itália (OSWALD, 2020). Mas foi apenas em 1921 que se deu a invenção do polígrafo (KHAN et al., 2021), o qual, diferentemente das máquinas que o precederam, adquiriu popularidade ao ser utilizado em tribunais como ferramenta científica capaz de gerar provas a serem consideradas seriamente em um julgamento (OSWALD, 2020).

Observável na Figura 1, o polígrafo é um equipamento que registra, em diferentes canais, as respostas fisiológicas de um indivíduo a determinadas perguntas, registrando o batimento cardíaco, duração da respiração, reações galvânicas na pele, pressão sistólica e diastólica (FIEDLER; SCHMID; STAHL, 2002).

Há diferentes versões do polígrafo, com diferentes técnicas para a elaboração de perguntas, todavia o mais popular é o CQT (*Control Question Test*, Teste da Pergunta de Controle) (FIEDLER;

Figura 1 – Administração do polígrafo. Imagem registrada na década de 1970.

Fonte: FBI (2022)

SCHMID; STAHL, 2002). Nele, 10 a 12 perguntas são selecionadas pelo especialista na máquina, todas devendo ser respondidas com "Sim" ou "Não". Dentre elas, algumas são irrelevantes e utilizadas apenas para aquecimento e checagem do equipamento, enquanto as outras são divididas entre perguntas de controle e perguntas críticas. As de controle são selecionadas de modo a aumentar o nervosismo do interrogado, mas cujas respostas são irrelevantes, a fim de registrar o padrão das respostas fisiológicas do indivíduo. Já as críticas são aquelas cuja resposta deseja-se classificar como verdade ou mentira, tendo as respostas fisiológicas do indivíduo a tais perguntas comparadas com as respostas das perguntas de controle. Caso o nível de nervosismo do indivíduo em uma pergunta crítica esteja abaixo do nível médio das perguntas de controle, considera-se que ele está falando a verdade. Caso tal nível esteja acima, considera-se que ele mente. Todavia, caso tais níveis estejam muito próximos, tem-se um resultado inconclusivo.

Fiedler, Schmid e Stahl (2002) levantam uma série de problemas metodológicos na utilização do polígrafo para DD, os quais foram considerados pela Suprema Corte Alemã para abandonar o uso de tal equipamento em contextos judiciais. Dentre tais problemas, destacamos a falta de objetividade do procedimento, o qual depende da atuação subjetiva do especialista tanto na elaboração das perguntas quanto na avaliação das respostas fisiológicas registradas pela máquina.

Além de falhas metodológicas que comprometem sua confiabilidade, o polígrafo pode ser facilmente burlado. Em estudo realizado por Honts, Raskin e Kircher (1994), após um treinamento de até 30 minutos sobre técnicas para burlar o polígrafo, cerca de 50% dos participantes puderam "vencer" a máquina, demonstrando a falta de eficácia do dispositivo no contexto de DD.

Com base nisso, nota-se que o polígrafo não cessou a busca de pesquisadores por métodos de automatizar DD.

3

Machine Learning na Detecção de Mentiras

O recente sucesso de ML na execução de um grande número de tarefas (LAI; TAN, 2019) despertou o interesse de pesquisadores para a aplicação de tais técnicas no campo de DD. Dada a baixa acurácia humana e a ineficácia do polígrafo, a aplicação de técnicas de ML em tal campo apresentou grande potencial. Nesse sentido, este capítulo discute a utilização de ML em DD, tanto em sistemas que utilizam de métodos não-visuais (Seção 3.1) quanto visuais (Seção 3.2). Ademais, as limitações de tal utilização são discutidas (Seção 3.3) bem como questões éticas que a envolvem (Seção 3.4).

3.1 Sistemas Não Baseados em Vídeo

Dentre sistemas de ML cujos dados utilizados não incluem vídeos, destacamos os sistemas que utilizam textos, áudios, EEG e fNIRS (*Functional Near-Infrared Spectroscopy*, Espectroscopia Funcional em Infravermelho Próximo).

Hernández-Castañeda et al. (2017) combinaram três *datasets* textuais e utilizam como classificadores *Support Vector Networks* (CORTES; VAPNIK, 1995) e *Naïve Bayes*, obtendo acurácia de 55,55%. Já Delgado et al. (2021) combinam diferentes técnicas de ML, dentre as quais ANN, atuando em um *dataset* que combina *fake news*, e-mails e notícias verídicas, alcançando acurácia de 82,35%. Ho e Hancock (2019), por sua vez, extraem *features* (características) de textos utilizados em CMC (Comunicação Mediada por Computador) classificando-os a partir de um modelo de regressão linear. Seus resultados, todavia, são apresentados a partir do erro médio quadrático, o que impossibilita a comparação direta com os demais trabalhos, que utilizam acurácia. Por outro lado, o trabalho de Ruiter e Kachergis (2018) não desenvolve um sistema de DD, mas foca na criação de um *dataset* textual de 9000 documentos, somando ao grande número de *datasets* textuais existentes na área de DD (HERNÁNDEZ-CASTAÑEDA et al., 2017).

Em contrapartida, a modalidade de áudio não apresenta um número expressivo de *datasets*

públicos. Assim, Xue, Rohde e Finkelstein (2019) criam seu próprio *dataset* a partir de gravações realizadas com voluntários em um jogo envolvendo verdades e mentiras. Tais gravações foram classificadas pela combinação de modelos de ML e DL por votação, sendo eles regressão linear, árvores de decisão, RF (*Random Forest*), *Gradient Boosting Classifier*, SVM (*Support Vector Machine*), *Stochastic Gradient Descent* e LSTM (*Long Short-Term Memory*), com acurácia de 55,8%. De modo similar, Mendels et al. (2017) criaram seu próprio *dataset* com cerca de 120 horas de gravações, as quais foram classificadas tanto a partir de regressão linear e RF quanto por modelos de DL como LSTM, obtendo F1 Score (KORSTANJE, 2021) de 63,9% para o modelo de DL e precisão de 76,11% para o RF. Já Gonzalez-Billandon et al. (2019) combinam *features* extraídas de áudios com pupilometria obtidas a partir de um experimento realizado com 28 participantes que gerou 510 mentiras e 504 verdades, as quais foram classificadas por RF com acurácia de 73%.

A utilização de EEG para DD se baseia no fato de que uma onda denominada P300 é emitida pelo cérebro quando um indivíduo se depara com um objeto familiar dentre um conjunto de objetos desconhecidos, de modo que a detecção de tal onda em uma situação em que o indivíduo alega não ter conhecimento de um objeto implica em uma mentira (DODIA et al., 2020). Nesse sentido, Dodia et al. (2020) criaram um *dataset* composto por testes realizados com 20 indivíduos, contendo 600 amostras, as quais foram classificadas por diferentes técnicas, mas obtendo acurácia máxima de 88,3% com uma ANN simples de uma única camada oculta. Já Baghel et al. (2020) utilizam o *dataset* criado por Gao et al. (2014), extraindo as informações dos EEGs para um vetor de características que é classificado por uma CNN (*Convolutional Neural Network*, Rede Neural Convolucional) unidimensional, obtendo acurácia de 82%. O mesmo *dataset* é utilizado por Amber et al. (2019), mas as informações dos EEGs são convertidas em uma imagem que é classificada por uma CNN bidimensional, alcançando a impressionante acurácia de 99,69%. Todavia, os próprios autores indicam a possibilidade de *overfitting* (sobreajuste) dado o tamanho reduzido do *dataset*, o qual contém dados de apenas 30 indivíduos.

Por sua vez, a utilização de fNIRS para DD parte do princípio de que o cérebro humano aumenta sua atividade durante a produção de mentiras, aumentando seu consumo de oxigênio, o que ocasiona uma mudança de concentração tanto de hemoglobina desoxigenada quanto de hemoglobina oxigenada, as quais podem ser detectadas a partir do espectro infravermelho e, portanto, de fNIRS (HERNANDEZ-REYNOSO; GARCIA-GONZALEZ, 2013). Nesse sentido, Hernandez-Reynoso e Garcia-Gonzalez (2013) realizaram um experimento com uma única voluntária, tendo suas respostas em um jogo envolvendo verdades e mentiras registradas por fNIRS, as quais foram então classificadas por uma ANN com acurácia de 83,33%.

Contudo, ainda que sistemas de ML a partir de EEG e fNIRS alcancem valores satisfatórios de acurácia, em geral maiores que sistemas baseados em texto ou áudio, sua utilização em contextos reais envolvendo mentiras de alto impacto, como interrogatórios, tribunais ou entrevistas, apresenta obstáculos dada a natureza da aquisição de tais sinais. As Figuras 2 e 3 apresentam os

dispositivos utilizados para tal tarefa, os quais, além de exigir conhecimento especializado para sua operação, não são itens facilmente encontrados no cotidiano, em contrapartida a microfones e câmeras.

Figura 2 – Dispositivo para realizar EEG.

Fonte: Expo (2022)

Figura 3 – Dispositivo para registrar fNIRS.

Fonte: BIOPAC (2022)

Já a modalidade textual para sistemas de ML para DD esbarra na necessidade de uma etapa prévia de transcrição quando utilizada em contextos de comunicação oral como tribunais e da possibilidade de falsificação de textos quando utilizada em CMC, como é o caso de avaliações de hotéis. Um indivíduo mal intencionado pode facilmente copiar uma avaliação negativa verdadeira, criando assim uma avaliação negativa mentirosa, mas cuja estrutura léxica é a mesma de uma avaliação verdadeira, sendo indistinguível a partir de um sistema de ML.

Por sua vez, sistemas baseados em áudio sofrem da escassez de *datasets* públicos para sua construção e aprimoramento, o que se comprova pela necessidade de criação de um *dataset* próprio por cada um dos trabalhos analisados (XUE; ROHDE; FINKELSTEIN, 2019; MENDELS et al., 2017; GONZALEZ-BILLANDON et al., 2019). Além disso, ainda que a modalidade de áudio alcance maior acurácia humana em relação à modalidade de vídeo (BOND; DEPAULO, 2006), isso não se observa em sistemas de ML que realizaram DD a partir de diferentes modalidades de dados, conforme Tabela 1, na qual se nota que a acurácia ou a AUC (*Area Under Curve*, Área Sob a Curva) (NARKHEDE, 2018) em vídeos é maior que em áudios.

3.2 Sistemas Baseados em Vídeo

Como apresentado na Tabela 1, sistemas de ML baseados em vídeo apresentam, em geral, melhores resultados para DD que outras modalidades de dados obtidos a partir de dispositivos de fácil acesso. Devido a isso, muitos trabalhos desenvolvem tal tipo de sistema, seja realizando a classificação exclusivamente a partir de vídeos ou realizando classificação multimodal combinando-os com textos, áudios ou mesmo outros tipos de dados.

Tabela 1 – Comparação das acurácias entre diferentes modalidades de dados para sistemas de ML para DD. Acurácias marcadas com * são, na realidade, AUC

AUTORES	TEXTO	ÁUDIO	VÍDEO
(KRISHNAMURTHY et al., 2018)	90,24%	52,38%	**93,08**%
(MATHUR; MATARIC, 2020)	63%	72%	**76**%
(JAISWAL; TABIBU; BAJPAI, 2016)	**67,2**%	34,23%	66,12%
(WU et al., 2018)	66,25%*	81,71%*	**89,88**%*

Fonte: Autor

Khan et al. (2021) criam um *dataset* a partir de entrevistas realizadas com 100 voluntários, as quais totalizaram 1.200 vídeos com tempos variando de 3 a 6 minutos que são classificados por RF com acurácia de 78%. Analogamente, Bhaskaran et al. (2011) criam um *dataset* a partir de experimento realizado com 40 indivíduos, extraindo *features* dos vídeos gerados e obtendo acurácia de 82,5% em um modelo bayesiano. Por sua vez, Soldner, Pérez-Rosas e Mihalcea (2019) reúnem 25 vídeos disponíveis no YouTube com média de 6 minutos do quadro "Box of Lies" do programa televisivo "The Tonight Show Starring Jimmy Fallon". Após extração de *features* visuais e auditivas dos vídeos, bem como de *features* geradas a partir da transcrição destes, a classificação é realizada por RF com acurácia de 69%.

Apesar de tais trabalhos criarem *datasets* próprios, a maioria dos trabalhos não o faz, visto que Pérez-Rosas et al. (2015) construíram o primeiro *dataset* público de mentiras de alto impacto, o RLT (*Real-Life Trial*), composto de 121 vídeos, dos quais 61 apresentam mentiras e 60 apresentam verdades. A duração média dos vídeos é de 28 segundos retratando 56 indivíduos distintos (21 do sexo feminino e 35 do sexo masculino) com idades variando de 16 a 60 anos. As transcrições de cada um dos vídeos e anotações indicando gestos e expressões faciais observados em cada um deles acompanham o *dataset*. A Figura 4 apresenta exemplos de quadros retirados dos vídeos utilizados. Dada a relevância de se realizar experimentos com mentiras de alto impacto (VRIJ, 2008), o RLT tornou-se o padrão para sistemas de ML em DD (MATHUR; MATARIC, 2020). Com base nisso, o Quadro 1 compara trabalhos de ML para DD que utilizam o RLT, os quais são aqui discutidos.

Apesar de relatar acurácia de 100%, o trabalho de Venkatesh et al. (2020) tem metodologia pouco detalhada e utiliza de técnicas pouco sofisticadas se comparado com os demais trabalhos, visto que utiliza uma ANN composta de CNN e LSTM, arquitetura que, conforme exposto na Seção 4.1, tende a apresentar resultados inferiores a outras técnicas. Trabalhos similares dos autores não foram encontrados, comprometendo a plena compreensão dos métodos por eles adotados para alcançar tamanha acurácia a partir de uma técnica simples e comprometendo, consequentemente, a credibilidade de seus resultados. Em contrapartida, Ding et al. (2019) descrevem detalhadamente seus métodos e utilizam de técnicas sofisticadas como GANs (*Generative Adversarial Networks*) (GOODFELLOW et al., 2014) e um *backbone* baseado na ResNet50 (HE et al., 2016), alcançado acurácia de 93,61% para a classificação exclusivamente

Figura 4 – Exemplos de quadros do RLT. Os dois quadros no topo são de vídeos demonstrando mentiras e os dois quadros abaixo representam verdades.

Fonte: Pérez-Rosas et al. (2015)

Quadro 1 – Comparação entre trabalhos que implementam sistemas de ML para DD a partir do *dataset* RLT. São apresentadas as acurácias obtidas para a classificação exclusivamente a partir de vídeos (Acc. V) e para a combinação de vídeos, áudio e texto (Acc. V+A+T). Acurácias marcadas com * são, na realidade, AUC.

Autores	**Acc. V**	**Acc. V+A+T**	**Técnica**	**Observações**
(VENKATESH et al., 2020)	100	-	CNN + LSTM	Metodologia pouco detalhada
(DING et al., 2019)	93,61	97,00	CNN	ResNet, GANs
(KRISHNAMURTHY et al., 2018)	93,08	96,14	3DCNN	Fusão por produto de Hadamard
(GOGATE; ADEEL; HUSSAIN, 2017)	78,57	96,42	3DCNN	Fusão por concatenação
(NGÔ et al., 2018)	72,8	-	CNN + RNN	Reconstrução Facial
(CARISSIMI; BEYAN; MURINO, 2018)	99	99	SVM	*Features*: AlexNet, *Multiview Learning*
(WU et al., 2018)	89,88*	92,21*	Regressão Logística	*Features*: *Improved Dense Trajectory*
(AVOLA et al., 2019)	76,84	-	RBF-SVM	*Features*: *Action Units*
(MATHUR; MATARIC, 2020)	76	84	SVM	*Features*: OpenFace, *Affect*
(JAISWAL; TABIBU; BAJPAI, 2016)	67,2	78,95	SVM	*Features*: *Action Units*

Fonte: Autor

a partir de vídeos e 97% para a classificação multimodal. Conforme descrito pelos autores, a utilização de GANs remedia o tamanho limitado do *dataset* RLT, contribuindo para resultados

melhores.

A utilização de CNNs tridimensionais (3DCNNs) é realizada tanto por Krishnamurthy et al. (2018) quanto por Gogate, Adeel e Hussain (2017). Apesar desses trabalhos apresentarem diferença significativa na acurácia obtida a partir de vídeos, seu resultado para a combinação de vídeos, áudios e textos foi bastante similar. Somando-se às diferenças de tais trabalhos, o primeiro utilizou de produto Hadamard para realizar a fusão das *features* de cada modalidade de dado, enquanto o segundo utilizou de concatenação.

Utilizando de uma proposta distinta, Ngô et al. (2018) utilizou CNNs e RNNs (*Recursive Neural Networks*, Redes Neurais Recursivas) treinando-as para realizar a tarefa de reconstrução facial a partir dos vídeos do RLT. Após o treinamento do modelo para tal finalidade, o vetor de características gerado ao final é utilizado para alimentar uma RNN que efetua a classificação do vídeo em verdade ou mentira. Todavia, a acurácia obtida ficou em 72,8%, abaixo dos demais trabalhos.

Carissimi, Beyan e Murino (2018) combinam técnicas de ML com DL ao utilizarem a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2017) para realizar extração de características dos vídeos, as quais são então classificadas tanto a partir de *Multiview Learning* (XU; TAO; XU, 2013) quanto por uma SVM. Os resultados obtidos para *Multiview Learning* ficaram ligeiramente abaixo (98%) daqueles obtidos pela SVM (99%) para a classificação multimodal. Já a classificação exclusivamente por vídeo alcançou mesma acurácia da classificação multimodal, demonstrando a eficácia da utilização da AlexNet para extração de características.

Os demais trabalhos presentes no Quadro 1 focam em modelos de ML. Wu et al. (2018) utilizam *Improved Dense Trajectory* (WANG et al., 2016) como *features* com Regressão Logística, obtendo valores expressivos de AUC, mas não informando a acurácia obtida, dificultando a comparação de seus resultados com os demais trabalhos. Avola et al. (2019), por sua vez, utilizam *Action Units* (BALTRUŠAITIS; MAHMOUD; ROBINSON, 2015) como *features* e uma SVM com kernel de função de base radial (RBF-SVM) obtendo acurácia de 76,84%. Similarmente, Mathur e Mataric (2020) utilizam a OpenFace (BALTRUŠAITIS; ROBINSON; MORENCY, 2016) para determinar as *Action Units*, mas também utilizando dados sobre *Affect* (RUSSELL, 1980), alcançando resultados similares com uma SVM simples. Já Jaiswal, Tabibu e Bajpai (2016) utilizam meramente *Action Units* como *features* visuais, classificando-as numa SVM simples e obtendo acurácia aquém da dos demais trabalhos.

3.3 Limitações

Apesar de obter resultados promissores, sistemas de ML para DD apresentam importantes limitações no que diz respeito a seu desenvolvimento e a sua utilização em cenários reais.

Conforme discutido por Vrij (2008), mentiras de baixo impacto são significativamente

distintas de mentiras de alto impacto tanto no contexto na qual elas ocorrem quanto em seu conteúdo e na forma como são contadas, o que sugere a impossibilidade de que um sistema seja treinado a partir de um tipo mas utilizado para classificar o outro. Por um lado, o maior interesse no desenvolvimento de técnicas de DD se dá pelas potenciais consequências de mentiras de alto impacto mas, por outro, pode não ser possível registrar tal tipo de mentiras em experimentos laboratoriais (VRIJ, 2008).

A fim de potencializar o impacto das mentiras contadas em contexto laboratorial, alguns experimentos buscaram introduzir penalidades e/ou recompensas para os participantes encarregados de contar verdades e mentiras, dentre os quais se destaca um experimento no qual participantes cujas mentiras fossem identificadas deveriam ficar trancados em uma sala por uma hora enquanto ruídos de 110 decibéis eram sequencialmente disparados (MANN; VRIJ; BULL, 2004). Tal tipo de experimento é altamente questionável do ponto de vista ético e dificilmente seria aprovado atualmente. Ademais, códigos de ética atualmente em vigor e leis de proteção de dados internacionais limitam ainda mais a possibilidade de se introduzir mentiras de alto impacto em contextos laboratoriais, exigindo que estudos sejam realizados a partir de mentiras contadas em contextos reais.

Todavia, há uma significativa carência de *datasets* públicos envolvendo mentiras de alto impacto contadas em contextos reais com dados em formato de vídeo ou áudio. Para textos e EEG, tal carência não é tão profunda, contudo tais modalidades apresentam suas próprias limitações para serem desenvolvidas e utilizadas em contextos reais, as quais foram discutidas na Seção 3.1. Nesse sentido, os trabalhos de Fitzpatrick e Bachenko (2012) e Gokhman et al. (2012) abordam o processo de construção de *datasets* para DD, estimulando pesquisadores a tal empreita, dada a escassez de *datasets* públicos na área. Contrastando profundamente com os 121 vídeos que compõem o RLT, a área de classificação de imagens conta com *datasets* públicos como a ImageNet (DENG et al., 2009) com 14.197.122 imagens e a área de classificação de ações conta com o Sports 1-M (KARPATHY et al., 2014), com 1.133.158 vídeos.

3.4 Questões Éticas

Dada a atual inexistência de grandes *datasets* públicos para DD envolvendo mentiras de alto impacto contadas em contextos reais, a utilização de sistemas de ML em contextos reais de DD não é recomendada. Tendo em vista que um modelo de ML pode obter bons resultados para o conjunto de dados com os quais foi treinado e testado mas ser incapaz de generalizar para exemplos distintos daqueles com o qual já teve contato até então, a utilização de um modelo treinado em um *dataset* pequeno é acompanhada de grande incerteza da sua real eficácia.

Conforme discutido por Khan et al. (2021), a criação de um *dataset* deve ser suficientemente inclusiva dado seu contexto de utilização a fim de evitar qualquer tipo de discriminação. Nesse sentido, um *dataset* ideal para DD deve conter indivíduos contando verdades e/ou mentiras

de alto impacto em contextos reais, sendo tais indivíduos de diferentes gêneros, orientações sexuais, nacionalidades, etnias, idades, níveis de escolaridade e quaisquer outros recortes sociais relevantes.

Tendo em vista que tal *dataset* ideal ainda não existe, trabalhos como o de Lai e Tan (2019) e Kleinberg e Verschuere (2021) investigaram a utilização de sistemas híbridos, no qual a máquina realiza a classificação do dado de entrada entre verdade ou mentira e proporciona a uma pessoa informações para que esta possa realizar a classificação. Ambos os estudos concluíram que a acurácia é inversamente proporcional ao nível de participação do ser humano na decisão, sendo máxima quando a máquina a realiza independentemente. Todavia, remover a responsabilidade da máquina pode ser um aspecto desejável, especialmente em contextos de alta relevância como tribunais, permitindo também que ocorra certo nível de supervisão humana sobre a operação dos algoritmos.

Apesar das limitações e riscos associados, a utilização de sistemas de ML em DD apresenta o potencial de aumentar a acurácia humana na realização desta tarefa, requerendo que maior pesquisa seja realizada para o desenvolvimento da área. Ainda que potenciais erros gerados por tais sistemas sejam relevantes ao se considerar sua utilização ou mesmo seu desenvolvimento, cabe ressaltar que erros humanos em DD são frequentes. Gross e Shaffer (2012), em relatório sobre "inocentamentos" (tradução livre de *exonerations*, indicando casos em que um indivíduo encarcerado foi inocentado) realizados nos Estados Unidos entre 1989 e 2012, apontam que, dentre os 873 "inocentamentos" realizados, 53% (cerca de 462 ocorrências) envolvem mentiras em geral. Em outras palavras, são 462 casos de encarceramento indevido que poderiam ter sido evitados caso mentiras tivessem sido adequadamente classificadas.

4

Deep Learning Aplicado a Reconhecimento de Vídeos

O desenvolvimento de sistemas de DL baseados em vídeo esbarra na complexa natureza de dados em formato de vídeo. Eles são formados por quadros temporalmente distribuídos, os quais são compostos por pixels bidimensionalmente distribuídos, cada pixel contendo (para vídeos coloridos) um valor de intensidade para cada um dos canais RGB. Um vídeo de 30 segundos a 24 quadros por segundo com resolução de 640x360, por exemplo, é composto por 720 imagens nessa resolução, e tem, de forma bastante aproximada, complexidade computacional equivalente a um áudio de 8,64 horas com taxa de 128 kbit/s ou um texto de cerca de 138.240 páginas.

Nesse sentido, o presente capítulo discute a utilização de DL para reconhecimento de vídeos, tal que a Seção 4.1 discute diferentes tipos de arquiteturas utilizadas para essa tarefa, enquanto a Seção 4.2 compara arquiteturas consideradas estado da arte, indicando aquela que apresenta melhor equilíbrio entre desempenho e custo computacional para sua utilização, a qual tem seu funcionamento descrito na Seção 4.3.

4.1 Reconhecimento de Vídeos

Tendo em vista a natureza de dados em formato de vídeo, seu reconhecimento requer que tanto as informações espaciais (em cada quadro que compõe o vídeo) quanto as informações temporais (obtidas a partir da variação temporal entre os quadros do vídeo) sejam consideradas, a fim de que a real compreensão semântica do vídeo seja possibilitada. Supondo um vídeo hipotético que apresente uma pessoa chorando de rir, a classificação realizada a partir das informações espaciais de um único quadro cuidadosamente selecionado permitiria identificar que a pessoa está chorando, todavia é só a partir da combinação dos diferentes quadros e a obtenção das informações temporais que se torna possível indicar que a pessoa está chorando de rir, mas não por estar triste.

A partir dessa necessidade de se combinar informações espaciais e temporais, diferentes técnicas são utilizadas para a criação de arquiteturas de DL para reconhecimento de vídeos, sendo possível categorizar os diferentes modelos utilizados com base nas técnicas empregadas em sua construção. Diferentes trabalhos de revisão bibliográfica (XIAO; XU; WAN, 2016; ASADI-AGHBOLAGHI et al., 2017; HERATH; HARANDI; PORIKLI, 2017; WU et al., 2018; ZHANG et al., 2017; KHURANA; KUSHWAHA, 2018; REN et al., 2019; SHARMA, 2020; ZHU et al., 2020) categorizam de forma particular as principais arquiteturas estado da arte criadas para tal tarefa. Todavia, podemos combinar as classificações propostas em tais trabalhos e categorizar as diferentes arquiteturas de reconhecimento de vídeos em quatro principais categorias:

- **CNN + RNN**: Realizam o processamento das informações espaciais de cada quadro utilizando CNNs bidimensionais, para então utilizar os dados de saída obtidos em RNNs que processam as informações temporais. A maioria dos modelos nessa categoria utiliza LSTMs como RNNs. Pode ser considerada a categoria com implementação mais simples, mas que obtém os piores resultados, frequentemente sendo incapaz de superar técnicas tradicionais de processamento de imagens que não envolvem ML. Um exemplo de arquitetura que alcança bons resultados e se encaixa em tal modelo é a proposta por Ng et al. (2015).

- **Two-Stream**: Também realizam o processamento das informações espaciais a partir de CNNs bidimensionais. Todavia, aqui as informações temporais são obtidas a partir do OF (*Optical Flow*, Fluxo Ótico), o qual representa a movimentação dos pixels entre diferentes quadros do vídeo e deve ser calculado previamente ao treinamento do modelo. O OF é então processado por uma ANN paralela à que realiza o processamento dos quadros individuais, tal que o resultado de ambas as ANNs são combinados ao final. Todavia, o cálculo do OF pode ser computacionalmente custoso e exigir grande espaço de armazenamento, chegando a requerer 4,5 TB para o *dataset* Kinetics-400 (KAY et al., 2017), de acordo com Zhu et al. (2020). Devido a isso, alguns modelos nessa categoria efetuam o treinamento de ANNs para automaticamente calcularem o OF e utilizá-lo no restante da rede, eliminando a necessidade de calculá-lo previamente. O primeiro trabalho a propor tal categoria de arquitetura foi o de Simonyan e Zisserman (2014), enquanto trabalhos como o de Zhu et al. (2019) e Kwon et al. (2020) buscam realizar o cálculo automático do OF.

- **3DCNN**: A utilização de 3DCNNs permite que quadros de um vídeo sejam agrupados e processados conjuntamente pela ANN. Assim, o processamento espacial se dá simultaneamente ao temporal a partir de uma mesma CNN, a qual utiliza de filtros tridimensionais. Devido ao grande número de parâmetros treináveis de tais modelos, o custo computacional dos primeiros modelos desta categoria se mostrou significativamente maior que o das demais, alcançando, todavia, os maiores valores de acurácia. Contudo, o interesse em reduzir tamanho custo motivou o surgimento de modelos de 3DCNN que equilibram o

custo computacional com a alta performance, alcançando resultados ligeiramente inferiores aos modelos mais robustos mas com custo significativamente menor. O trabalho seminal para tal tipo de arquitetura, conforme Zhu et al. (2020), foi o de Ji et al. (2010). Todavia, outros trabalhos como o de Tran et al. (2019) e Feichtenhofer et al. (2019) também se encaixam em tal categoria e buscam trazer maior eficiência nas arquiteturas propostas.

- **Inovadores**: A busca por ANNs capazes de realizar classificação de vídeos com baixo custo computacional motivou o surgimento de modelos que divergem das categorias supracitadas, não utilizando OF nem 3DCNNs, mas que alcançam resultados próximos a tais categorias com custos significativamente inferiores. Tais modelos utilizam CNNs bidimensionais mas extraem a componente temporal dos vídeos a partir de técnicas específicas de cada modelo e que frequentemente definem sua nomenclatura. Como exemplo, temos a arquitetura TSM, cujo nome é derivado da técnica proposta por seus autores para extrair a componente temporal dos vídeos, a *Temporal Shift Module* (LIN; GAN; HAN, 2019).

4.2 Arquiteturas Estado da Arte

A seleção de uma arquitetura pra reconhecimento de vídeos deve levar em conta não apenas sua acurácia, mas também o custo computacional por ela exigido para seu treinamento e posterior classificação de amostras. Ainda que, conforme discutido na Seção 4.1, modelos de 3DCNNs alcancem os melhores resultados, seu custo elevado pode inviabilizar a sua utilização ao se considerar limitações nos recursos físicos disponíveis. Conforme Zhu et al. (2020), o treinamento de uma 3DCNN no *dataset* Kinetics-400 utilizando 8 placas de vídeo de ponta pode levar 10 dias de execução ininterrupta, valor que pode ser extrapolado para meses de execução caso considere-se uma máquina com uma única placa de vídeo.

Nesse sentido, selecionamos arquiteturas consideradas estado da arte que foram desenvolvidas buscando equilibrar o custo computacional com a acurácia, o que levou à construção da Tabela 2. Nela, o custo computacional de cada modelo é dado a partir de GFLOPS/*view*, que indica quantos bilhões de operações de ponto flutuante devem ser realizadas para processar cada *view*, sendo uma *view* correspondente a um quadro utilizado para se obter informações espaciais de um vídeo. Tal distinção conceitual é realizada pois alguns modelos, como STM e SlowFast, utilizam apenas um subconjunto dos quadros de um vídeo para extrair as informações espaciais. Além disso, todas as arquiteturas Two-Stream listadas realizam cálculo do OF a partir da própria ANN.

As arquiteturas presentes na Tabela 2 foram agrupadas conforme seu custo, podendo-se observar que, para cada um dos grupos considerados, a SlowFast, em suas diferentes configurações, apresenta a melhor acurácia. Apesar de ser uma arquitetura do tipo 3DCNN, sua construção faz com que ela alcance expressivos resultados com baixo custo computacional. Tendo isso em vista e tomando como critério o equilíbrio entre desempenho da arquitetura e o custo computacional

Tabela 2 – Comparação de arquiteturas consideradas estado da arte para reconhecimento de vídeos. Os dados de acurácia e custo computacional foram obtidos a partir dos trabalhos indicados na coluna Fonte(s), sendo a acurácia referente ao desempenho da arquitetura no *dataset* Kinetics-400. A repetição de arquiteturas na tabela se dá por diferenças de configuração destas, gerando diferentes resultados para um mesmo modelo. As arquiteturas são divididas em quatro grupos com base em seu custo computacional, destacando-se aquela com maior acurácia de cada grupo.

ARQUITETURA	CATEGORIA	CUSTO (GFLOPS/*view*)	ACURÁCIA (%)	AUTOR(ES)	FONTE(S)
Hidden TSN	Two-Stream	-	72,8	(ZHU et al., 2019)	(ZHU et al., 2020)
TSM	Inovador	33	74,1	(LIN; GAN; HAN, 2019)	(LI et al., 2020)
TEINet	Inovador	33	74,9	(LIU et al., 2020)	(LIU et al., 2020)
MSNet	Two-Stream	34	75,0	(KWON et al., 2020)	(KWON et al., 2020)
TEA	Inovador	35	75,0	(LI et al., 2020)	(LI et al., 2020)
Slowfast ResNet50	**3DCNN**	**36,1**	**75,6**	**(FEICHTENHOFER et al., 2019)**	**(LIU et al., 2020)**
TSM	Inovador	65	74,7	(LIN; GAN; HAN, 2019)	(LI et al., 2020), (LIU et al., 2020)
Slowfast ResNet50	**3DCNN**	**65,7**	**77,0**	**(FEICHTENHOFER et al., 2019)**	**(FEICHTENHOFER et al., 2019)**
TEINet	Inovador	66	76,2	(LIU et al., 2020)	(LIU et al., 2020)
STM	Inovador	67	73,7	(JIANG et al., 2019)	(LI et al., 2020)
MSNet	Two-Stream	67	76,4	(KWON et al., 2020)	(KWON et al., 2020)
TEA	Inovador	70	76,1	(LI et al., 2020)	(LI et al., 2020)
CSN	3DCNN	73,8	76,2	(TRAN et al., 2019)	(TRAN et al., 2019)
CSN	3DCNN	83	76,7	(TRAN et al., 2019)	(TRAN et al., 2019)
CSN	3DCNN	96,7	76,8	(TRAN et al., 2019)	(TRAN et al., 2019)
Slowfast ResNet101	**3DCNN**	**106**	**77,9**	**(FEICHTENHOFER et al., 2019)**	**(LIU et al., 2020)**
CSN	3DCNN	108,8	77,8	(TRAN et al., 2019)	(TRAN et al., 2019)
NL Slowfast ResNet101	**3DCNN**	**234**	**79,8**	**(FEICHTENHOFER et al., 2019)**	**(LI et al., 2020), (LIU et al., 2020)**

Fonte: Autor

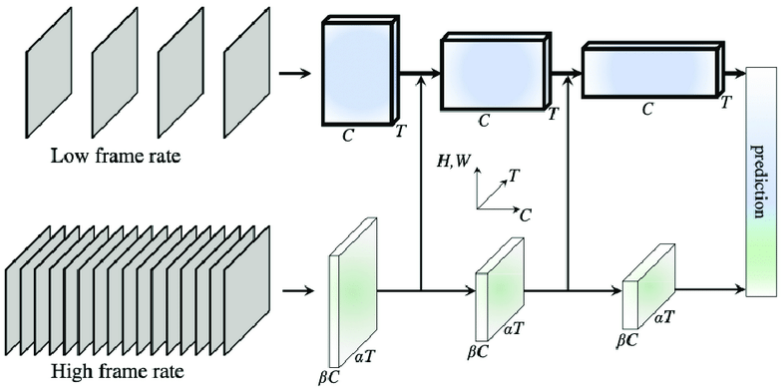
de sua utilização, a SlowFast se mostra a arquitetura mais indicada para trabalhos que realizem reconhecimento de vídeos e tenham limitações no poder computacional disponível.

4.3 SlowFast

A arquitetura SlowFast, desenvolvida por Feichtenhofer et al. (2019), é assim nomeada por ser composta de um modelo que combina uma via rápida e uma via lenta (Figura 5). A via lenta é responsável por obter a informação espacial dos vídeos, utilizando uma taxa de quadros baixa e muitos canais convolucionais, enquanto a via rápida é responsável por obter a informação temporal, utilizando uma taxa de quadros alta e poucos canais convolucionais. Tal combinação é parcialmente inspirada em estudos biológicos de células visuais de primatas, em que cerca de 15 a 20% das células operam em alta frequência mas têm baixa sensibilidade a cores ou detalhes espaciais, enquanto cerca de 80% das células operam em baixa frequência mas proporcionam grande detalhamento visual. Coincidentemente, cerca de 20% dos parâmetros treináveis da rede se localizam na via rápida, enquanto os demais compõem a via lenta.

A via lenta da SlowFast pode ser utilizada a partir de diferentes *backbones*, tal que a Tabela 2 apresenta seu resultado para os *backbones* ResNet50, ResNet101 e ResNet101 com

Figura 5 – Arquitetura SlowFast. A via superior tem baixa taxa de quadros (*Low frame rate*), sendo a via lenta, enquanto a via inferior tem alta taxa de quadros (*High frame rate*) e é a via rápida. C representa o tamanho dos canais utilizados na CNN enquanto T representa a resolução temporal. α e β são hiperparâmetros da via rápida que alteram, respectivamente, sua taxa de amostragem temporal e o processamento realizado sob cada quadro. Ao final da rede, ambas as vias são combinadas para se chegar à predição (*prediction*).



Fonte: Feichtenhofer et al. (2019)

blocos NL (*Non-Local*) (WANG et al., 2018). Além disso, o número de quadros utilizados na via lenta a cada iteração (τ) e o espaçamento temporal entre tais quadros (T) pode variar, de modo que as diferentes configurações da SlowFast são descritas em termos de $\tau \times T$ e podem levar a diferentes custos e resultados mesmo que utilizem um mesmo *backbone*. Ainda que a Tabela 2 apresente duas acurácias para a SlowFast com *backbone* ResNet50, a versão com menor custo tem configuração 4x16 (4 quadros utilizados na via lenta a cada iteração, com espaçamento de 16 quadros entre cada um deles) enquanto a versão com maior custo tem configuração 8x8.

5 Metodologia

O presente trabalho visa construir uma ANN para DD e, com base nos Capítulos 2, 3 e 4, utiliza o *dataset* RLT e a arquitetura SlowFast para sua construção. Tendo isso em vista, o presente capítulo aborda as metodologias utilizadas para tal implementação, descrevendo as ferramentas utilizadas na Seção 5.1, o processo de preparação do RLT na Seção 5.2, a separação dos vídeos do *dataset* na Seção 5.3 e os hiperparâmetros selecionados para treinamento na Seção 5.4.

O código completo da implementação está disponível no GitHub (CAMARA, 2022a).

5.1 Ferramentas utilizadas

A principal ferramenta utilizada no presente trabalho foi o *framework* GluonCV (GUO et al., 2020), o qual encapsula algoritmos de visão computacional abstraindo a complexidade de código inerente a eles, possibilitando acesso a versões otimizadas de tais métodos e facilitando a rápida implementação de arquiteturas estado da arte de ANNs. O GluonCV utiliza como base o Apache MXNet (FOUNDATION, 2022) para realizar o treinamento de ANNs, bem como funções de carregamento e separação de *dataset*.

Também foram utilizados o Pandas (MCKINNEY, 2010), biblioteca de análise de dados da linguagem de programação Python, e o TensorBoard, kit de visualização do TensorFlow (ABADI et al., 2015). O primeiro permitiu a leitura, a partir de arquivos CSV, de informações referentes ao *dataset* relevantes à separação dos vídeos em diferentes conjuntos, além de possibilitar o armazenamento em arquivo dos resultados das execuções durante o processo de treino e validação da ANN. Já o segundo, permitiu a geração e visualização de gráficos indicativos da acurácia do modelo nos conjuntos de treino e validação em tempo real durante o treinamento da ANN.

Além de tais ferramentas, o editor de vídeo Kdenlive (KDENLIVE, 2022) foi utilizado na etapa de preparação do *dataset* descrita na Seção 5.2.

A máquina utilizada para os experimentos realizados conta com CPU Intel Core i3-10100F 3,6 GHz, Memória RAM 16 GB 2.666 MHz, GPU GeForce GTX 1650 4 GB com sistema operacional Ubuntu 20.04 LTS. O código foi executado a partir do terminal integrado do editor de código-fonte Visual Studio Code (MICROSOFT, 2022).

5.2 Preparação do *dataset*

O *dataset* RLT foi obtido a partir da página pessoal de uma de suas autoras (MIHALCEA, 2016).

De modo semelhante ao realizado por Ngô et al. (2018), Ding et al. (2019), Wu et al. (2018), Mathur e Mataric (2020), Jaiswal, Tabibu e Bajpai (2016) e Carissimi, Beyan e Murino (2018), uma seleção prévia dos vídeos do RLT foi realizada, descartando-se os vídeos considerados inapropriados para a tarefa de DD. Para tanto, consideramos como inadequados todos os vídeos nos quais a face do indivíduo contando a verdade ou mentira estava oculta, fora de foco durante a maior parte do vídeo ou mesmo que contavam com múltiplos indivíduos em primeiro plano, dificultando, de um ponto de vista meramente visual, a determinação de qual indivíduo estaria contando a verdade ou mentindo. Dentre os vídeos remanescentes, aqueles que apresentavam trechos de baixa qualidade foram editados visando alcançar melhores resultados no treinamento da ANN. O processo de seleção dos vídeos válidos e da edição dos vídeos com trechos insatisfatórios é descrito em maiores detalhes no Apêndice A.

Após a etapa de seleção, 11 vídeos foram eliminados do RLT, tal que ele passou a ter 110 vídeos, 53 representando mentiras e 57 representando verdades. Tais vídeos retratam 51 indivíduos distintos, os quais apresentam diferentes quantidades de vídeos. Tendo em vista tal distribuição desequilibrada, o Quadro 2 relaciona a frequência de indivíduos para as diferentes quantidades de vídeos de verdades e mentiras. Destaca-se, em tal Quadro, que 41 indivíduos apresentam um único vídeo no *dataset* (13 com apenas uma mentira e 28 com apenas uma verdade), de modo que cerca de 37% dos vídeos correspondem a cerca de 80% dos indivíduos presentes no RLT. A relação detalhada entre os vídeos que retratam cada um dos indivíduos é apresentada no Apêndice B.

5.3 Separação do *dataset*

A separação do RLT em conjuntos de treino e teste foi realizada por Ding et al. (2019), Krishnamurthy et al. (2018) e Wu et al. (2018) mediante a técnica k-*Fold* (BROWNLEE, 2018b), a qual separa os vídeos do *dataset* em k conjuntos disjuntos. Por k iterações, $k-1$ conjuntos são utilizados para treino e o outro conjunto (o qual muda a cada iteração) é utilizado para teste, tomando-se a média da acurácia de teste de tais iterações como o valor da acurácia de teste para a ANN. Ao utilizar, iterativamente, diferentes vídeos para treino e teste, a k-*Fold* minimiza o efeito

Quadro 2 – Relação entre a frequência de indivíduos para diferentes quantidades de vídeos no *dataset* RLT após a etapa de seleção de vídeos.

Frequência de indivíduos	**Total de vídeos**	**Mentiras**	**Verdades**
1	21	18	3
1	12	7	5
1	9	6	3
1	8	0	8
1	7	5	2
1	4	4	0
4	2	0	2
13	1	1	0
28	1	0	1

Fonte: Autor

da aleatoriedade da separação dos conjuntos, chegando a um valor de acurácia representativo da real taxa de acerto da ANN para amostras que lhe são inéditas. Todavia, tal técnica utiliza de busca exaustiva de todas as combinações possíveis de hiperparâmetros para cada uma de suas iterações, tornando-se inviável para o presente trabalho em virtude das limitações da máquina utilizada para treinamento da ANN.

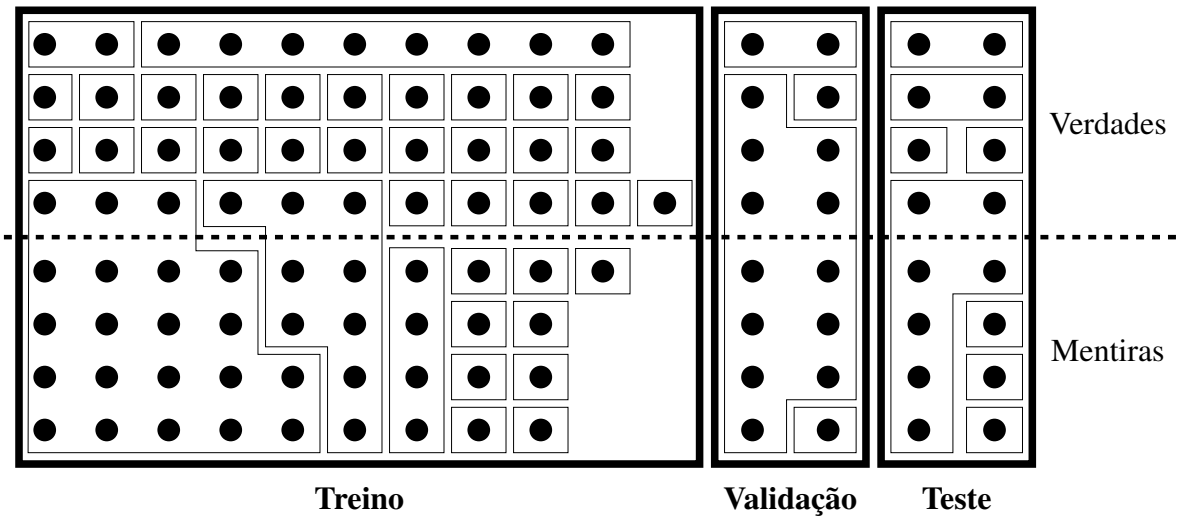
Tendo isso em vista e buscando simultaneamente otimizar o processo de busca dos hiperparâmetros ideais e determinar de forma precisa a capacidade de generalização da ANN a partir de sua acurácia de teste, inicialmente duas separações do RLT foram realizadas: a separação A, composta por conjuntos de treino, validação e teste, a qual foi utilizada na etapa de busca dos hiperparâmetros; e a separação B, que utilizou a técnica de 5-*Fold* e foi aplicada após a busca de hiperparâmetros para as 5 combinações que apresentaram melhores resultados na etapa prévia. Após a realização de testes na separação B e de sua análise, notou-se a necessidade de efetuar uma terceira separação, denominada separação C, a qual também utilizou da técnica 5-*Fold*, mas distribuindo as amostras dos 10 indivíduos com múltiplos vídeos no *dataset* entre as diferentes *folds*.

Para a separação A, o RLT foi dividido entre os conjuntos de treino, validação e teste seguindo a proporção 70/15/15, com 70% dos vídeos compondo o conjunto de treino, 15% o conjunto de validação e 15% o conjunto de teste. Dado que 15% de 110 resulta em valor fracionário, decidimos utilizar 16 vídeos para o conjunto de validação, com 8 verdades e 8 mentiras, e o mesmo número para o conjunto de teste, com os 78 vídeos remanescentes sendo alocados ao conjunto de treino.

Todavia, tal separação não se deu de forma totalmente aleatória. Conforme realizado por Ding et al. (2019), Krishnamurthy et al. (2018) e Wu et al. (2018), efetuamos a separação dos indivíduos presentes no *dataset* entre os diferentes conjuntos. Ou seja, a separação dos conjuntos foi realizada de modo que todos os vídeos de um mesmo indivíduo estivessem em um mesmo

conjunto, sem que houvesse a separação dos vídeos de um indivíduo entre os diferentes conjuntos, sendo tal método ilustrado na Figura 6.

Figura 6 – Possível separação dos vídeos conforme separação A. Cada círculo representa um vídeo, com as caixas indicando os diferentes indivíduos. Destaca-se que todos os indivíduos têm seus vídeos restritos a um único conjunto.



Fonte: Autor

A partir disso, um *script*, cujo pseudocódigo é apresentado no Código 1, foi criado para efetuar a separação dos vídeos do RLT nos conjuntos de treino, validação e teste, mas respeitando a restrição supracitada. O *script* seleciona aleatoriamente um indivíduo dentre os 51 que compõem o *dataset* reduzido, verificando se o conjunto de validação comporta os vídeos de tal indivíduo, considerando o limite máximo de 8 mentiras e 8 verdades para tal conjunto. Caso não comporte, o conjunto de teste é então verificado, sendo os vídeos do indivíduo alocados para o conjunto de treino caso nenhum dos demais conjuntos possam comportá-los. Tal processo se repete até que todos os indivíduos sejam alocados para um dos conjuntos. Entretanto, caso o sorteio dos indivíduos inviabilize o preenchimento dos conjuntos de validação e teste, o processo é reiniciado até que todos os conjuntos sejam preenchidos conforme desejado.

Já para a separação B, as mesmas considerações utilizadas na separação A foram observadas, distribuindo-se os vídeos do RLT entre 5 diferentes conjuntos, chamados de "*folds*", conforme a técnica 5-*Fold*. A separação B utilizou de algoritmo similar à separação A, efetuando a distribuição dos vídeos entre os conjuntos a partir dos indivíduos retratados no *dataset*. Todavia, aqui a separação foi realizada visando criar 5 conjuntos de 22 vídeos cada. Devido ao desbalanceamento no número de verdades e mentiras por indivíduo no RLT, não foi possível garantir que cada *fold* fosse composta de um mesmo número de mentiras e verdades, sendo a distribuição de tais categorias para cada *fold* representada na Tabela 3.

A separação C, tal qual a separação B, utilizou da técnica 5-*Fold*. Todavia, nela os vídeos de indivíduos com múltiplos vídeos no *dataset* foram distribuídos entre as *folds* visando balancear

Código 1 – Pseudocódigo do *script* de construção da separação A, a qual divide o *dataset* RLT entre conjuntos de treino, validação e teste.

```
individuos_alocados = 0
while individuos_alocados != 51:
    individuo_sorteado = sorteia_individuo()
    if validacao_comporta(individuo_sorteado):
        aloca_validacao(individuo_sorteado)
    elif teste_comporta(individuo_sorteado):
        aloca_teste(individuo_sorteado)
    else:
        aloca_treino(individuo_sorteado)
    individuos_alocados += 1
    if individuos_alocados==51 and validacao.num_vids+teste.num_vids<32:
        individuos_alocados = 0
        limpa(validacao)
        limpa(teste)
        limpa(treino)
```

Tabela 3 – Separação do *dataset* RLT conforme separação B. Cada *fold* conta com 22 vídeos e diferentes *folds* não apresentam vídeos de um mesmo indivíduo.

Conjunto	**Mentiras**	**Verdades**
Fold 0	18	4
Fold 1	11	11
Fold 2	11	11
Fold 3	11	11
Fold 4	2	20

Fonte: Autor

o número de vídeos de tais indivíduos entre os diferentes conjuntos, além de balancear o número de verdades e mentiras em cada *fold*, conforme pode ser observado na Tabela 4.

Tabela 4 – Separação do *dataset* RLT conforme separação C em 5 *folds* balanceadas. Cada *fold* conta com 22 vídeos e indivíduos com múltiplos vídeos no *dataset* têm seus vídeos distribuídos de forma balanceada entre as diferentes *folds*.

Conjunto	**Mentiras**	**Verdades**
Fold 0	11	11
Fold 1	11	11
Fold 2	11	11
Fold 3	11	11
Fold 4	9	13

Fonte: Autor

Tal separação foi realizada de acordo com o pseudocódigo apresentado no Código 2, no qual uma lista dos indivíduos com múltiplos vídeos no *dataset* é utilizada (*indiv_multiplos_videos*), ordenada de forma decrescente a partir do número de vídeos de cada indivíduo (conforme Apêndice B). Assim, os indivíduos com maior número de vídeos têm seus vídeos atribuídos às *folds* por

primeiro, sendo os indivíduos com apenas um vídeo atribuídos de forma aleatória posteriormente. Para cada indivíduo selecionado, uma *fold* é sorteada, verificando-se se ela comporta novos vídeos respeitando as quantidades máximas de amostras de cada categoria apresentadas na Tabela 4. Caso tal *fold* não comporte o vídeo, a próxima *fold* é selecionada, considerando-se percurso circular pela lista de *folds*. O processo se repete até que todos os indivíduos do *dataset* tenham seus vídeos alocados.

Código 2 – Pseudocódigo do *script* de construção da separação C, a qual divide o *dataset* RLT em 5 *folds* balanceadas.

```
indiv_alocados = 0
indiv_multiplos_videos = [3, 2, 1, 22, 7, 12, 25, 30, 36, 37]
indice_indivs = 0
while indiv_alocados != 51:
    if indice_indivs<10:
        indiv_selecionado = indiv_multiplos_videos[indice_indivs]
        indice_indivs += 1
    else:
        indiv_selecionado = sorteia_indiv()

    indice_fold = sorteia_fold()
    for video in videos_indiv(indiv_selecionado):
        while not fold[indice_fold].comporta(video):
            indice_fold = (indice_fold+1)%5
        aloca_video(indice_fold)
        indice_fold = (indice_fold+1)%5

    indiv_alocados += 1
```

Um exemplo que ilustra as diferentes estratégias das separações B e C é dado na Tabela 5, a qual apresenta o número de vídeos por *fold* do indivíduo 3, o qual conta com o maior número de vídeos do *dataset*. Enquanto na separação B todos os seus vídeos foram alocados à *Fold* 0, na separação C eles foram distribuídos buscando equilibrar, tanto quanto possível, o número de seus vídeos para cada *fold*. Nota-se que, para tal exemplo, a *fold* sorteada na construção da separação C foi a *Fold* 3, com os demais vídeos sendo alocados iterativamente a partir dela.

Tabela 5 – Número de vídeos por *fold* para o indivíduo 3 conforme separações B e C.

	Separação B	**Separação C**
***Fold* 0**	21	4
***Fold* 1**	0	4
***Fold* 2**	0	4
***Fold* 3**	0	5
***Fold* 4**	0	4

Fonte: Autor

5.4 Hiperparâmetros de treinamento

Tomando como base os trabalhos de Ding et al. (2019), Ngô et al. (2018), Carissimi, Beyan e Murino (2018) e do trabalho que apresenta a rede SlowFast (FEICHTENHOFER et al., 2019), os seguintes hiperparâmetros de treinamento foram avaliados com os seguintes valores:

- Taxa de aprendizagem: $5 * 10^{-3}$, 10^{-3}, $5 * 10^{-4}$, 10^{-4};
- Estratégia de decaimento da taxa de aprendizagem:
  - Sem decaimento;
  - Decaimento a cada 10 épocas até a época 100;
  - Decaimento a cada 10 épocas até a época 200;
  - Decaimento nas épocas 40, 80 e 100;
- Otimizador: Adam, SGD (*Stochastic Gradient Descent*) e RMSProp;
- Número de épocas: 100 e 200;
- *Weight decay*: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4};
- Configurações da SlowFast:
  - 4x16 ResNet50 sem pré-treino;
  - 4x16 ResNet50 com pré-treino no Kinetics-400 (KAY et al., 2017);
  - 8x8 ResNet50 com pré-treino no Kinetics-400.

A estratégia de decaimento na taxa de aprendizagem foi inspirada por Feichtenhofer et al. (2019) e Ding et al. (2019) e diz respeito à redução da taxa de aprendizagem em um fator de 10 em épocas pré-determinadas. Enquanto os primeiros autores utilizam uma função matemática para determinar tais épocas, os segundos a realizam a cada 10 iterações. Nesse sentido, optamos por avaliar a utilização de uma estratégia de decaimento similar à de Ding et al. (2019) comparada com uma estratégia de decaimento mais esparsa (com decaimento apenas nas épocas 40, 80 e 100). Além disso, avaliamos o comportamento da ANN caso nenhuma estratégia de decaimento fosse utilizada.

Já a SlowFast, conforme descrito na Seção 4.3, pode ser utilizada a partir de diferentes configurações e *backbones*, de modo que optamos por utilizar as diferentes configurações da arquitetura disponíveis no GluonCV. Dadas as limitações da máquina utilizada para treinamento, apenas as configurações com *backbone* ResNet50 foram avaliadas, sendo duas delas pré-treinadas no Kinetics-400 e outra sem pré-treino.

Além dos hiperparâmetros citados, tem-se o *batch size*, cujo valor utilizado foi de 1, em virtude das limitações de poder computacional da máquina utilizada.

6

Resultados e Discussão

Com base na metodologia descrita no Capítulo 5, o presente capítulo apresenta e discute os resultados obtidos pela execução de tais etapas. A Seção 6.1 apresenta e discute os valores encontrados na etapa de busca de hiperparâmetros (Seção 5.4) a partir da separação A. Em seguida, a Seção 6.2 apresenta e analisa os resultados encontrados pela implementação da ANN a partir da separação B. A Seção 6.3, por sua vez, apresenta e discute os resultados encontrados na realização de testes utilizando a separação C. Por fim, a Seção 6.4 aponta o resultado final da implementação da ANN, discutindo as diferentes separações utilizadas e comparando o valor obtido com aqueles encontrados por trabalhos correlatos.

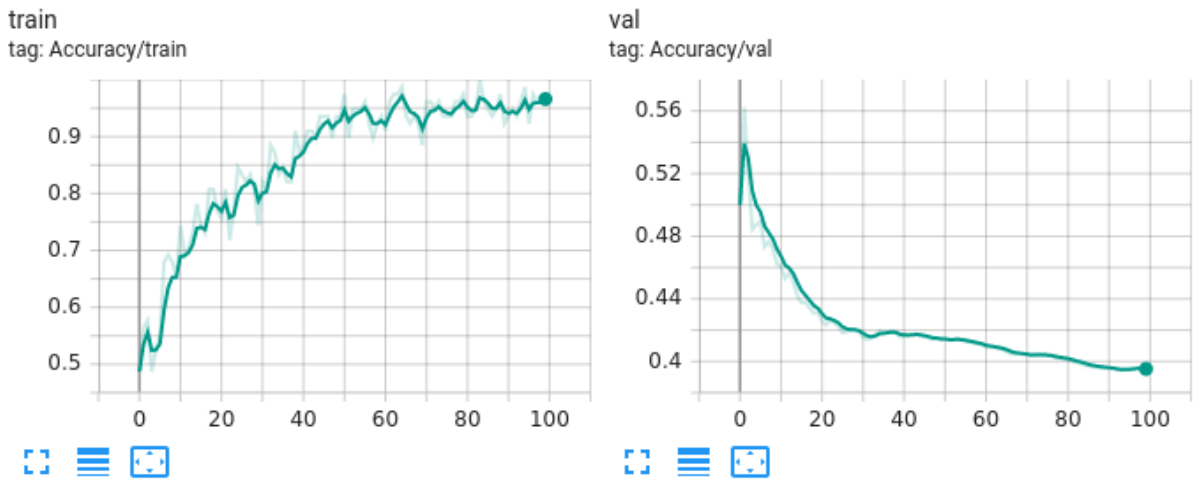
Por utilizar de aleatoriedade em sua construção e a fim de proporcionar reprodutibilidade experimental, as separações A, B e C aqui utilizadas têm sua constituição apresentada no Apêndice C, em que os indivíduos presentes em cada conjunto são relacionados.

6.1 Busca por Hiperparâmetros

A fim de encontrar a combinação de hiperparâmetros ideal, 69 execuções foram realizadas a partir da separação A (descrita na Seção 5.3), cada uma utilizando combinações únicas dos hiperparâmetros descritos na Seção 5.4. Para cada uma, avaliou-se a acurácia no conjunto de treino e validação, sendo tais valores apresentados no Apêndice D. Além disso, os gráficos com as variações das acurácias de treino e validação ao longo do treinamento podem ser encontrados no TensorBoard (CAMARA, 2022b), sendo alguns deles aqui reproduzidos.

Dentre as 69 execuções realizadas, o valor médio da diferença entre a acurácia de treino e de validação (diferença a qual denominaremos "valor de *overfitting*") foi de 29%, indicando forte tendência a *overfitting*. Tal diferença chegou à marca de 57,94% para a execução de ID 52 (a qual é representada na Figura 7), tendo valor inferior a 10% para apenas 8 execuções (11,6% do total de execuções).

Figura 7 – Exemplo de *overfitting* em uma das execuções realizadas na busca pelos hiperparâmetros ideais. Nota-se que a ANN continuou a melhorar sua acurácia no conjunto de treino ao longo do treinamento, em detrimento de sua acurácia no conjunto de validação e, consequentemente, de sua capacidade de generalização.



Fonte: Autor

No que diz respeito aos resultados observados nas 69 execuções, apenas 18 apresentaram acurácia de validação superior a 54% (valor referente à acurácia humana para DD, conforme discutido na Seção 2.2), indicando o impacto negativo do *overfitting* na capacidade de generalização da ANN.

Tal tendência a *overfitting* pode ser justificada pela utilização da arquitetura SlowFast, a qual foi desenvolvida especialmente para *datasets* como o Charades (SIGURDSSON et al., 2016), o qual conta com 203 categorias de classificação e 9848 vídeos, o Kinetics-400 (KAY et al., 2017), com 400 categorias e 306.245 vídeos, e o Kinetics-600 (CARREIRA et al., 2018), com 600 categorias e 495.547 vídeos, sendo tais *datasets* consideravelmente mais complexos que o RLT, que contou apenas com 2 categorias de classificação e 110 vídeos após a etapa de seleção. Ainda que a SlowFast alcance resultados superiores a arquiteturas de reconhecimento de vídeos com custo similar no Kinetics-400 (conforme discutido na Seção 4.2), foi possível observar que sua complexidade faz com que ela não seja a mais adequada a *datasets* pequenos como o RLT.

Quanto aos hiperparâmetros utilizados, foi possível identificar padrões entre eles a partir da análise do comportamento dos gráficos gerados no TensorBoard para as acurácias de treino e validação ao longo das épocas de cada execução, bem como da comparação dos resultados de cada execução com outras que utilizaram de hiperparâmetros similares.

De forma distinta dos demais hiperparâmetros, os testes realizados com o *weight decay* não geraram informações conclusivas a respeito da influência de seus diferentes valores sobre o treinamento da ANN. Nesse sentido, o valor de 10^{-4} foi favorecido, seguindo o padrão utilizado

por Feichtenhofer et al. (2019).

Para a taxa de aprendizagem, o valor de 0,0001 se mostrou demasiadamente pequeno, visto que o melhor resultado encontrado para tal valor foi de 55,44% para a acurácia de validação. Os demais valores avaliados alcançaram resultados superiores a 59%, não havendo distinção significativa entre sua eficácia, de modo que todos foram considerados aceitáveis para análises futuras.

Já para as diferentes estratégias de decaimento da taxa de aprendizagem, testes foram realizados comparando as estratégias A (sem decaimento), B (com decaimento a cada 10 épocas até a época 100) e D (com decaimento nas épocas 40, 80 e 100) com diferentes valores de taxa de aprendizagem. A Tabela 6 resume os resultados de tais testes. Nela, nota-se que a estratégia A se mostrou inadequada, enquanto a estratégia B se mostrou promissora. Já a estratégia D, apesar de obter média da acurácia de validação abaixo da acurácia humana, alcançou a acurácia de 59,88% para uma das execuções (ID 6), mostrando-se promissora ao ser utilizada com taxa de aprendizagem de 0,005. A diferença nos resultados de cada estratégia de decaimento pode ser justificada por uma maior influência de *overfitting* à medida que as diferentes estratégias resultam em taxas de aprendizagem maiores ao final do treinamento da ANN. Quanto maior o número de decaimentos, menor o valor da taxa de aprendizagem ao final da execução e, para as estratégias avaliadas, menor o valor do *overfitting*.

Tabela 6 – Comparação entre as estratégias de decaimento da taxa de aprendizagem para execuções de até 100 épocas. Acc. Val. representa a acurácia obtida no conjunto de validação e Núm. Decaimentos diz respeito a quantos decaimentos foram realizados entre o início do treinamento e o final da época 99.

Estratégia	**Média Acc. Val.**	**Média Valor *Overfitting***	**Núm. Decaimentos**
A	49%	32,3%	0
B	55,8%	5%	9
D	51,6%	22,7%	2

Fonte: Autor

Posteriormente, considerando execuções de 200 épocas, as estratégias de decaimento B e C (com decaimento a cada 10 épocas até a época 200) foram comparadas, com um melhor resultado sendo alcançado pela estratégia B, a qual obteve acurácia de validação média de 53,9% com uma das execuções atingindo a marca de 61,38%, enquanto a estratégia C obteve média de 50,8%. Tendo isso em vista, a estratégia B se mostrou a mais recomendada.

A avaliação dos otimizadores se deu mediante realização de múltiplas execuções com cada otimizador, variando os demais hiperparâmetros entre cada execução, levando em conta os resultados e análises descritos previamente. Tomando as execuções que utilizaram o otimizador Adam com as execuções equivalentes do RMSProp e SGD, tem-se os resultados apresentados na Tabela 7. A partir dela, é possível notar que o otimizador SGD alcançou melhores resultados que os demais otimizadores, sendo cerca de 4% melhor (relativamente) que os demais ao se considerar

todas as execuções, mas sendo cerca de 6% melhor (relativamente) ao se considerar apenas os três melhores resultados. Tal melhoria pode ser justificada por um valor de *overfitting* menor do SGD, o qual foi de aproximadamente metade do valor obtido pelos demais otimizadores.

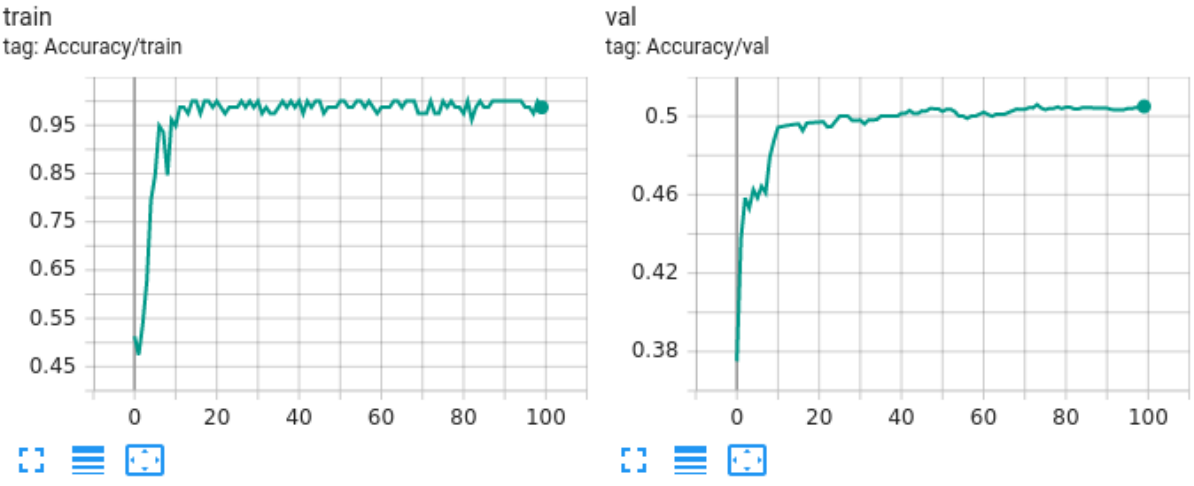
Tabela 7 – Comparação entre os otimizadores utilizados. Acc. Val. representa a acurácia obtida no conjunto de validação e Média Acc. Val. Top-3 indica a média de tal acurácia para os três melhores resultados.

Otimizador	**Média Acc. Val.**	**Média Acc. Val. Top-3**	**Média Valor *Overfitting***
Adam	49,41%	56,63%	34,25%
RMSProp	50,55%	56,02%	33,95%
SGD	51,93%	59,8%	17,94%

Fonte: Autor

Dentre as diferentes configurações da SlowFast avaliadas, notou-se que a realização de pré-treino no Kinetics-400 teve impacto significativo na capacidade de generalização da ANN. A Figura 8 apresenta as acurácias de treino e validação ao longo do treinamento da ANN utilizando a configuração 4x16 (4x16 ResNet50 sem pré-treino), sendo possível observar, no gráfico da acurácia de validação, que a ANN não melhora expressivamente sua capacidade de generalização ao longo do treino. Tal comportamento pode ser justificado observando-se o gráfico da acurácia de treino, em que, já na época 15, a ANN alcança acurácia de 100%, indicando *overfitting* elevado.

Figura 8 – Exemplo de execução com utilização da SlowFast sem pré-treino no Kinetics-400.



Fonte: Autor

A comparação da configuração 4x16 com a 4x16K (4x16 ResNet50 com pré-treino no Kinetics-400) e a 8x8K (8x8 ResNet50 com pré-treino no Kinetics-400) pode ser observada na Tabela 8, na qual, tanto para a média da acurácia de validação quanto para a média do valor de *overfitting*, a configuração 4x16K tem os melhores resultados, enquanto a 4x16 tem os piores.

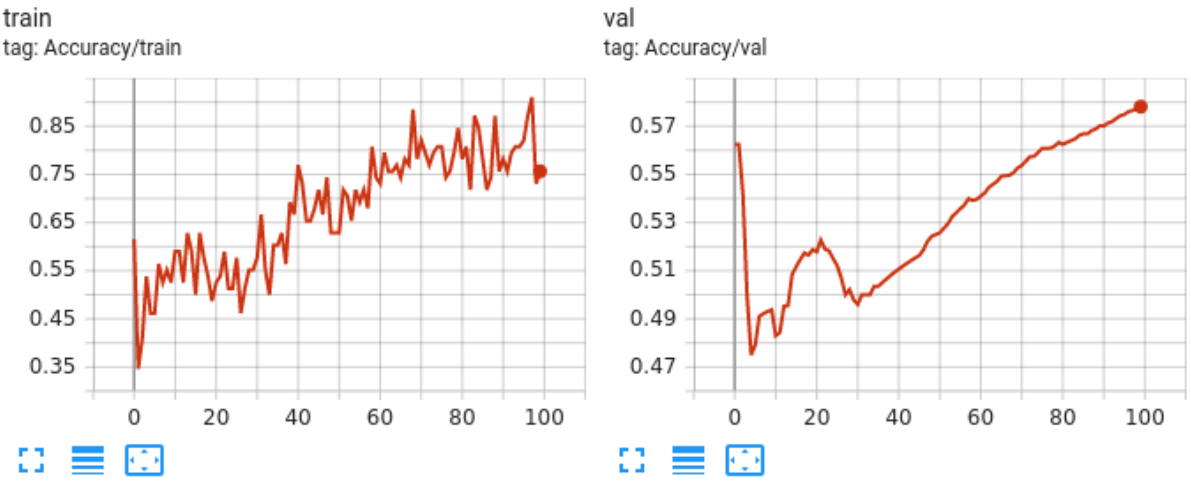
Tabela 8 – Comparação entre as diferentes configurações da SlowFast. Acc. Val. representa a acurácia obtida no conjunto de validação. A descrição de cada configuração é apresentada no texto.

Configuração da SlowFast	**Média Acc. Val.**	**Média Valor *Overfitting***
4x16K	59,8%	6,44%
4x16	50,62%	45,96%
8x8K	52,54%	20,11%

Fonte: Autor

Das 69 execuções, realizadas, as primeiras 63 utilizaram um número de épocas igual a 100, a fim de viabilizar a análise dos demais hiperparâmetros, tendo em vista que o tempo de treinamento da ANN é diretamente proporcional ao número de épocas utilizado. Com base nas análises previamente descritas, observou-se o crescimento da acurácia de validação nas últimas 20 épocas de treino, extrapolando-se tal crescimento para 200 épocas. A partir de tal extrapolação, todas as combinações de hiperparâmetros cuja acurácia de validação na época 200 superaria 60% foram executadas com 200 épocas. Assim, foi possível determinar o real potencial de combinações de hiperparâmetros que apresentaram alta tendência de crescimento na acurácia de validação em suas últimas 20 épocas, como é o caso da execução de ID 30, representada na Figura 9, bem como de combinações com baixa tendência de crescimento, mas que apresentaram resultados elevados.

Figura 9 – Exemplo de uma execução com tendência de crescimento para sua acurácia de validação. Extrapolando o crescimento observado nas últimas 20 épocas, uma execução de 200 épocas alcançaria acurácia de validação de 65,2%.



Fonte: Autor

6.2 Testes na Separação B

A partir das análises apresentadas e do resultado de cada uma das execuções realizadas a partir da separação A (descrita na Seção 5.3), efetuou-se a execução sobre a separação B das 5 combinações de hiperparâmetros com melhores resultados, sendo elas as combinações de IDs 6, 8, 11, 65 e 67, apresentadas no Quadro 3. Todas elas usam o otimizador SGD, a configuração 4x16 da Slowfast com pré-treino no *dataset* Kinetics-400 e *weight decay* de 0,0001.

Quadro 3 – Melhores combinações de hiperparâmetros encontradas a partir da separação A.

ID	Taxa de Aprendizagem	Estratégia de Decaimento da Taxa de Aprendizagem	Épocas	Acurácia de Validação
6	0,005	Épocas 40, 80 e 100	100	59,88%
8	0,001	A cada 10 épocas até a época 100	100	59,13%
11	0,0005	A cada 10 épocas até a época 100	100	60,38%
65	0,001	A cada 10 épocas até a época 100	200	61,38%
67	0,005	Épocas 40, 80 e 100	200	60,41%

Os resultados das execuções das combinações de hiperparâmetros apresentadas no Quadro 3 são apresentados na Tabela 9, enquanto as matrizes de confusão de cada teste realizado são apresentadas no Apêndice E.

Tabela 9 – Resultados percentuais da execução da técnica 5-*Fold* para a separação B a partir das 5 melhores combinações de hiperparâmetros obtidas na etapa de busca de hiperparâmetros. DP representa o Desvio Padrão em cada *fold* e N. Indiv. indica o número de indivíduos para cada *fold*, conforme Apêndice C.

ID	*Fold* 0	*Fold 1*	*Fold* 2	*Fold* 3	*Fold* 4	Média Final
6	9,09	54,55	54,55	45,45	22,73	37,27
8	90,91	59,09	45,45	36,36	68,18	60
11	90,91	45,45	68,18	22,73	68,18	59,09
65	50	59,09	40,91	22,73	50	44,55
67	18,18	50	59,09	18,18	40,91	37,27
Média	51,82	53,63	53,63	29,1	49,99	47,63
DP	38,78	5,93	10,85	11,42	19,29	17,25
N. Indiv.	2	10	10	7	22	10,2

Fonte: Autor

Pode-se notar, na Tabela 9, que a combinação de hiperparâmetros de ID 8 alcançou o melhor resultado dentre todas, com uma acurácia final de 60%. A análise do desempenho de tal combinação sobre as diferentes *folds* aponta a variabilidade gerada por divisões distintas do *dataset*, tal que seu resultado variou de 36,36% na *Fold* 3 a 90,91% na *Fold* 0.

No que diz respeito a possíveis tendências de classificação da ANN para uma das categorias, pode-se observar, na Tabela 10, que não houve tendência acentuada para nenhuma delas, sendo cerca de 54% dos vídeos classificados como mentiras e 46% como verdades.

Conforme pode ser observado no Apêndice E, as demais combinações de hiperparâmetros também não apresentaram tendência significativa de classificação, com o total de vídeos classificados como mentiras variando de 40 a 60%.

Tabela 10 – Matriz de confusão resultante para a combinação de hiperparâmetros de ID 8 a partir de testes realizados na separação B. M indica Mentira e V indica Verdade.

ID 8		**Saída**		
		M	V	Total
Classificação Real	M	34 (30,9%)	19 (17,3%)	53
	V	25 (22,7%)	32 (29,1%)	57
	Total	59	51	110

Fonte: Autor

A partir das matrizes de confusão de cada teste realizado, foi possível notar que, para indivíduos com um grande desbalanceamento do seu número de vídeos de verdades e mentiras, a ANN tende a classificar em uma mesma categoria a maioria dos vídeos da categoria predominante, impactando significativamente a acurácia de testes que têm tais indivíduos no conjunto de teste. Tal situação foi observada tanto na *Fold* 0 quanto na *Fold* 3.

A *Fold* 0 contém vídeos do indivíduo 3 (Apêndice C), o qual apresenta 18 mentiras e 3 verdades (Apêndice B). Conforme pode ser observado na Tabela 11, todas as execuções agruparam a maioria das 18 mentiras de tal indivíduo em uma mesma categoria. Visto que 18 vídeos correspondem a cerca de 82% dos vídeos da *fold*, tal agrupamento resultou, nos casos extremos, em acurácias de 9,09% e 90,91%, as quais são, respectivamente, a menor e a maior acurácia obtidas nos testes realizados (Tabela 9). Tendo tal comportamento em vista, testes realizados sobre a *Fold* 0, a qual conta com apenas 2 indivíduos, apresentaram um desvio padrão bastante elevado.

Tabela 11 – Impacto do indivíduo 3 sobre testes realizados na *Fold* 0 da separação B. Dentre os 22 vídeos da *Fold* 0, 18 são mentiras que retratam o indivíduo 3. Nota-se que, para a maioria das execuções realizadas, a maioria de tais vídeos recebeu uma mesma classificação.

ID	**Mentiras corretamente classificadas**	**Mentiras incorretamente classificadas**	**Acurácia**
6	0	18	9,09%
8	18	0	90,91%
11	16	2	90,91%
65	7	11	50%
67	1	17	18,18%

Fonte: Autor

Por sua vez, a *Fold* 3 contém vídeos do indivíduo 22 (Apêndice C), o qual apresenta 8 verdades e nenhuma mentira (Apêndice B). Conforme Tabela 12, a maioria dos vídeos de tal indivíduo foi classificada sob uma mesma categoria em todos os testes realizados, impactando na acurácia resultante de cada teste realizado sobre tal *fold*, visto que 8 vídeos correspondem a cerca de 36% dos vídeos da *fold*. Todavia, ao contar com 7 indivíduos, a *Fold* 3 apresentou desvio padrão menor que a *Fold* 0, com resultados mais uniformes.

Tabela 12 – Impacto do indivíduo 22 sobre testes realizados na *Fold* 3 da separação B. Dentre os 22 vídeos da *Fold* 3, 8 são verdades que retratam o indivíduo 22. Nota-se que, para a maioria das execuções realizadas, a maioria de tais vídeos recebeu uma mesma classificação.

ID	Verdades corretamente classificadas	Verdades incorretamente classificadas	Acurácia
6	8	3	45,5%
8	7	4	36,36%
11	3	8	22,73%
65	4	7	22,73%
67	2	9	18,18%

Fonte: Autor

Tal análise aponta uma consequência negativa do desbalanceamento elevado de categorias para um mesmo indivíduo no *dataset*, de modo que um *dataset* ideal construído para DD deva ter um balanceamento mais adequado de categorias para cada indivíduo nele retratado.

Outro aspecto negativo observado no RLT que impactou os resultados encontrados foi o desbalanceamento do número de vídeos por indivíduo. Conforme apresentado na Seção 5.2, dentre os 51 indivíduos presentes no *dataset*, 41 contam com apenas um vídeo, tal que 20% dos indivíduos representam 63% dos vídeos. Tal característica do RLT levou ao desbalanceamento da *Fold* 4, a qual conta com 22 indivíduos, cada um com um único vídeo na *fold*. Devido a isso, os treinamentos que utilizaram tal *fold* como teste contaram com apenas 29 dos 51 indivíduos (57%), alcançando a segunda menor acurácia média entre as *folds* e o segundo maior desvio padrão.

Por outro lado, a *Fold* 1 e a *Fold* 2, sendo menos afetadas por desbalanceamentos dos dados, apresentaram as melhores médias de acurácia e os menores desvios padrão.

6.3 Testes na Separação C

A análise do impacto negativo do desbalanceamento dos dados suscitou questionamentos acerca da regra de separação dos vídeos utilizada por Ding et al. (2019), Krishnamurthy et al. (2018) e Wu et al. (2018), em que todos os vídeos de um indivíduo devem ser agrupados em um mesmo conjunto. A partir disso, a separação C foi realizada, visando minimizar os efeitos do desbalanceamento dos dados do RLT.

A partir disso, as combinações de hiperparâmetros de IDs 6, 8, 11, 65 e 67 (conforme Apêndice D) foram novamente executadas mediante técnica 5-*Fold*, todavia agora sobre a separação C. As matrizes de confusão de tais execuções são apresentadas no Apêndice F e seus resultados, na Tabela 13, na qual pode-se observar que a combinação de ID 8 alcançou o melhor resultado, com acurácia de 66,36%.

Tabela 13 – Resultados percentuais da execução da técnica 5-*Fold* para a separação C. DP representa o Desvio Padrão em cada *fold* e N. Indiv. indica o número de indivíduos para cada *fold*, conforme Apêndice C.

ID	***Fold* 0**	***Fold* 1**	***Fold* 2**	***Fold* 3**	***Fold* 4**	**Média Final**
6	50	54,55	72,73	59,09	59,09	59,09
8	59,09	63,64	72,73	54,55	81,82	66,36
11	59,09	54,55	50	68,18	72,73	60,91
65	59,09	40,91	68,18	59,09	72,73	60
67	36,36	50	40,91	50	68,18	49,09
Média	52,73	52,73	60,91	58,18	70,91	59,09
DP	9,96	8,26	14,59	6,74	8,26	9,56
N. Indiv.	15	16	16	15	16	15,6

Fonte: Autor

Tal qual observado na separação B, aqui não houve tendência acentuada de classificação para uma das categorias, com o percentual de vídeos classificados como mentiras, para cada combinação testada, variando de 40 a 60% (conforme Apêndice F). Para a combinação de hiperparâmetros de ID 8, pode-se observar, na Tabela 14, a qual apresenta sua matriz de confusão para todas as *folds*, que o percentual de vídeos classificados como mentiras foi de 47,3%. Além disso, pode-se observar em tal tabela que a taxa de acerto para mentiras foi de 64,2% (34 acertos dentre as 53 mentiras no *dataset*) e de 68,4% para verdades (39 acertos dentre as 57 verdades no *dataset*), o que corrobora a inexistência de um forte viés de classificação para uma das categorias.

Tabela 14 – Matriz de confusão resultante para a combinação de hiperparâmetros de ID 8 a partir de testes realizados na separação C. M indica Mentira e V indica Verdade.

ID 8		**Saída**		
		M	V	Total
Classificação Real	M	34 (30,9%)	19 (17,3%)	53
	V	18 (22,7%)	39 (29,1%)	57
	Total	52	58	110

Fonte: Autor

A Tabela 15 compara os resultados obtidos pela execução da técnica 5-*Fold* sobre a separação B e sobre a separação C. Nota-se que, no que diz respeito à acurácia final, a

separação C obteve resultados melhores que a separação B, indicando uma influência positiva do balanceamento dos vídeos. O desvio padrão apresentou uma faixa de valores mais restrita na separação balanceada, com um valor médio de quase metade do valor obtido pela separação não balanceada. O balanceamento realizado gerou uma diferença na média do número de indivíduos presentes em cada *fold*, tal que na separação C um indivíduo pode figurar em múltiplas *folds*, sendo tal valor cerca de 50% maior em relação ao valor encontrado pela separação B.

Tabela 15 – Comparação entre resultados de testes realizados nas separações B e C. Acc. indica a acurácia, DP indica o Desvio Padrão e N. Indiv., o número de indivíduos que compõem cada *fold*. A observação entre parênteses ao lado dos piores e melhores casos indica em que contexto ocorreram.

	Separação B	**Separação C**
Pior Acc. Final	37,27 (ID 6)	49,09 (ID 67)
Melhor Acc. Final	60 (ID 8)	66,36 (ID 8)
Média Acc. Final	47,63	59,09
Pior DP	38,78 (*Fold* 0)	14,59 (*Fold* 2)
Melhor DP	5,93 (*Fold* 1)	6,74 (*Fold* 3)
Média DP	17,25	9,56
Média N. Indiv.	10,2	15,6

Fonte: Autor

6.4 Resultado Final

A utilização de três métodos de separação do *dataset* RLT permitiu que diferentes experimentos e análises pudessem ser realizadas, tal que cada separação serviu um propósito distinto.

A separação A viabilizou a etapa de busca de hiperparâmetros descrita na Seção 6.1, de modo que os diferentes valores de hiperparâmetros puderam ser comparados e analisados ao serem utilizados pela arquitetura SlowFast com o RLT. Todavia, sua utilização para realização de testes se mostrou problemática por depender fortemente da aleatoriedade da divisão entre os conjuntos de treino, validação e teste. Meramente por critério de comparação, tal técnica de separação foi utilizada para testes, mas seus resultados variaram de 12,5% a 87,5% de acordo com as diferentes distribuições realizadas, mostrando-se inadequada para verificar a real capacidade de generalização da ANN.

A utilização da separação A para a busca de hiperparâmetros e da separação B para a etapa de testes resultou num total de 10.900 épocas utilizadas para treinar diferentes configurações da ANN e obter-se sua acurácia. Dado que cada época levou, em média, 28 segundos para ser executada, tem-se um total de aproximadamente 85 horas ou 3,5 dias. Caso a técnica 5-*Fold* fosse utilizada na busca de hiperparâmetros com busca exaustiva para cada *fold* a partir da separação B, um total de 172.800 épocas seria necessário, totalizando cerca de 1.344 horas ou 56 dias. Ainda

que a utilização da 5-*Fold* garanta a combinação ótima dos hiperparâmetros e o melhor resultado para a acurácia da ANN, nota-se a inviabilidade de sua utilização a partir do *hardware* disponível para o presente trabalho.

A separação B, por sua vez, por utilizar a técnica 5-*Fold*, mostrou menor variabilidade de resultados, não sendo tão impactada pela aleatoriedade da divisão dos conjuntos e se mostrando mais propícia para a realização de testes. Conforme Tabela 9, os valores finais de acurácia obtidos variaram de 37,27% a 60%. Todavia, conforme análises descritas na Seção 6.2, notou-se que tal estratégia de separação do *dataset* era negativamente impactada pelo desbalanceamento de dados do RLT.

Com base nisso, a separação C se mostrou menos impactada por tal desbalanceamento. Conforme Tabela 13, seus resultados variaram de 49,09% a 66,36%, uma variação menor que as demais estratégias de separação. Como discutido na Seção 6.3, tal separação apresentou resultados melhores que a separação B, sendo portanto considerada aquela que melhor aponta a real acurácia da ANN e sua capacidade de generalização.

Ao se considerar o Quadro 1, a menor acurácia alcançada por trabalhos correlatos ao utilizar-se exclusivamente a modalidade de vídeo foi de 67,2%, a qual ainda é superior ao valor de 66,36% alcançado. Todavia, conforme observado previamente, a complexidade da arquitetura SlowFast frente ao tamanho reduzido do *dataset* RLT a torna altamente propensa a *overfitting*, prejudicando o resultado final. Nesse sentido, a acurácia final obtida pode ser considerada satisfatória frente às limitações enfrentadas.

7

Conclusão

O presente trabalho realizou a implementação de uma ANN profunda para DD com base em dados em formato de vídeo com acurácia de 66,36%, a qual é superior à acurácia humana de 54%.

Para tanto, uma revisão bibliográfica foi realizada tratando de questões centrais a tal tarefa, como o conceito de mentira (Seção 2.1), a acurácia humana em DD (Seção 2.2) e técnicas historicamente utilizadas para DD (Seção 2.3). A partir da análise e comparação de trabalhos que realizam DD a partir de ML (Seções 3.1 e 3.2), foi possível selecionar um *dataset* contendo vídeos de humanos mentindo e falando a verdade (Seção 3.2), o RLT, além de se investigar as limitações (Seção 3.3) e implicações éticas (Seção 3.4) da utilização de tal tipo de sistema em contextos reais.

A partir de uma revisão acerca de arquiteturas de DL utilizadas para reconhecimento de vídeos (Capítulo 4), a arquitetura SlowFast foi selecionada, sendo utilizada com o *dataset* RLT mediante uma série de ajustes e adaptações (Capítulo 5). As diferentes etapas de implementação foram então descritas e analisadas (Capítulo 6), chegando-se ao resultado final de 66,36%. Tal resultado foi obtido utilizando-se a configuração 4x16 da SlowFast com pré-treino no *dataset* Kinetics-400 (KAY et al., 2017), taxa de aprendizagem $0,001$, decaimento da taxa de aprendizagem a cada 10 épocas, otimizador SGD, *weight decay* de 0,0001 e 100 épocas de treinamento.

Tendo isso em vista, pode-se dizer que tanto o objetivo principal quanto os objetivos específicos do trabalho foram alcançados satisfatoriamente. Assim, com base nas etapas realizadas, o presente capítulo apresenta, na Seção 7.1, considerações sobre as técnicas e ferramentas utilizadas e discute, na Seção 7.2, direcionamentos a potenciais trabalhos futuros que utilizem o presente trabalho como base.

7.1 Considerações

A partir dos experimentos realizados, pudemos observar uma série de problemas e limitações no *dataset* RLT:

- Dentre os 121 vídeos do *dataset*, 11 deles se mostraram, por motivos diversos apresentados no Apêndice A, inviáveis à tarefa de DD, o que corresponde a 9,1% dos dados do RLT;
- Dos 110 vídeos restantes, 25 tiveram de ser editados (conforme Apêndice A) por apresentar trechos considerados inadequados à tarefa a ser executada pela ANN, o que corresponde a 22,7% do *dataset* reduzido;
- Conforme apresentado na Seção 5.2 e discutido na Seção 6.2, o RLT apresenta grande desbalanceamento de seus dados, tendo em vista que, dentre os 51 indivíduos presentes no *dataset* reduzido, 41 deles contam com apenas um vídeo e correspondem a apenas 37,3% do total de vídeos. Além disso, dentre os indivíduos que apresentam múltiplos vídeos, nenhum deles conta com um mesmo número de mentiras e verdades, havendo também um desbalanceamento entre as categorias para cada indivíduo.

Apesar de tais falhas, o RLT é considerado, até a data de escrita do presente trabalho, o *dataset* padrão para DD com ML a partir de vídeos. Ele é utilizado pela maioria dos trabalhos correlatos analisados pois, até onde se tem conhecimento, nenhum outro *dataset* público de vídeos que apresente mentiras de alto impacto em contexto real tenha sido criado desde o ano de sua publicação, em 2015.

Por se tratar de um *dataset* pequeno em relação a outros *datasets* utilizados para classificação de vídeos, o RLT alcançou resultado inferior ao ser classificado por uma arquitetura robusta como a SlowFast se comparado com os resultados obtidos por trabalhos correlatos. Todavia, a arquitetura SlowFast ainda alcançou resultado superior à acurácia humana no presente trabalho e se mostrou satisfatória na tarefa de classificação de vídeos, contando com diferentes configurações que lhe garantem maior desempenho em troca de poder computacional para seu treinamento e execução.

Por outro lado, o GluonCV, apesar de simplificar o processo de implementação de arquiteturas de visão computacional consideradas estado da arte e otimizar etapas de seu treinamento, limita significativamente a realização de modificações em tais arquiteturas e o acesso a dados detalhados a respeito do treinamento realizado. Devido a isso, não foi possível testar técnicas de redução de *overfitting* como *dropout* (BROWNLEE, 2018a) ou obter informações detalhadas sobre a classificação realizada pela ANN, como o percentual de acerto por indivíduo do *dataset*. Somando-se aos problemas do GluonCV, sua documentação é escassa (em relação a outros *frameworks* de DL), com muitas páginas contendo informações desatualizadas e, consequentemente, incorretas. Devido a isso e considerando o atual estado de tal *framework*, não

recomendamos sua utilização, sendo preferível o uso de *frameworks* com maior documentação e utilização, como o PyTorch (PASZKE et al., 2019).

7.2 Trabalhos Futuros

Reforçamos, com base nas discussões apresentadas na Seção 3.4, nos resultados obtidos e nas considerações apontadas a respeito do RLT, que a utilização de sistemas de ML para DD em contextos reais **não é recomendada**. Ainda que técnicas de ML e, em especial, DL, tenham se desenvolvido consideravelmente nos últimos anos, a maior carência atualmente enfrentada pela área de DD a partir de ML situa-se na disponibilidade de *datasets*.

Nesse sentido, recomendamos que trabalhos futuros foquem em tal carência, visando construir, a partir de dados públicos, *datasets* com humanos mentindo e falando verdades com dados em formato de vídeo, áudio e texto.

A construção de tal *dataset* envolve questões já investigadas pela literatura, como fatores éticos e estatísticos da construção de um *dataset* para ML, além de questões inéditas, como a comparação entre mentiras de alto e baixo impacto do ponto de vista de sistemas de ML.

O estudo de tais fatores na construção de um *dataset* de DD se faz altamente relevante visto que a aquisição de dados deve levar em conta diferentes grupos sociais, evitar desbalanceamentos de dados que possam comprometer a classificação realizada, bem como garantir exemplos de número suficiente para que sistemas que utilizem de tal *dataset* tenham capacidade de generalização suficientemente satisfatória para sua eventual utilização em contextos reais.

Já quanto à comparação entre mentiras de alto e baixo impacto, ainda que a literatura acerca de DD indique que tais categorias de mentiras são drasticamente distintas, nenhum trabalho de ML encontrado investigou a diferença entre tais categorias do ponto de vista de ML, apontando se um sistema treinado a partir de mentiras de baixo impacto poderia ser utilizado para classificar mentiras de alto impacto e, em última análise, se tais categorias efetivamente apresentam distinção significativa ao serem classificadas por modelos de ML.

Referências

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2015. Disponível em: <http://download.tensorflow.org/paper/whitepaper2015.pdf>. Citado na página 32.

AMBER, F. et al. P300 based deception detection using convolutional neural network. In: *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*. [S.l.: s.n.], 2019. p. 201–204. Citado na página 20.

ASADI-AGHBOLAGHI, M. et al. Deep learning for action and gesture recognition in image sequences: A survey. In: ____. *Gesture Recognition*. Cham: Springer International Publishing, 2017. p. 539–578. ISBN 978-3-319-57021-1. Disponível em: <https://doi.org/10.1007/978-3-319-57021-1_19>. Citado na página 28.

AVOLA, D. et al. Automatic deception detection in rgb videos using facial action units. In: *Proceedings of the 13th International Conference on Distributed Smart Cameras*. New York, NY, USA: Association for Computing Machinery, 2019. (ICDSC 2019). ISBN 9781450371896. Disponível em: <https://doi.org/10.1145/3349801.3349806>. Citado 2 vezes nas páginas 23 e 24.

BAGHEL, N. et al. Truth identification from EEG signal by using convolution neural network: Lie detection. In: *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. [S.l.: s.n.], 2020. p. 550–553. Citado na página 20.

BALTRUŠAITIS, T.; MAHMOUD, M.; ROBINSON, P. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In: *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. [S.l.: s.n.], 2015. v. 06, p. 1–6. Citado na página 24.

BALTRUŠAITIS, T.; ROBINSON, P.; MORENCY, L.-P. OpenFace: An open source facial behavior analysis toolkit. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2016. p. 1–10. Citado na página 24.

BHASKARAN, N. et al. Lie to me: Deceit detection via online behavioral learning. In: *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG)*. [S.l.: s.n.], 2011. p. 24–29. Citado na página 22.

BIOPAC. *fNIR FAQ*. 2022. Disponível em: <https://www.biopac.com/knowledge-base/fnir-faq/>. Acesso em: 25/06/2022. Citado na página 21.

BOND, C.; DEPAULO, B. Accuracy of deception judgments. *Personality and social psychology review: an official journal of the Society for Personality and Social Psychology, Inc*, v. 10, p. 214–34, 02 2006. Citado 3 vezes nas páginas 16, 17 e 21.

BROWNLEE, J. *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*. 2018. Disponível em: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>. Acesso em: 06/07/2022. Citado na página 51.

BROWNLEE, J. *A Gentle Introduction to k-fold Cross-Validation*. 2018. Disponível em: <https://machinelearningmastery.com/k-fold-cross-validation/>. Acesso em: 25/06/2022. Citado na página 33.

CAMARA, M. K. *SlowFastDeceptionDetection*. 2022. Disponível em: <https://github.com/MahatKC/SlowFastDeceptionDetection>. Acesso em: 25/06/2022. Citado na página 32.

CAMARA, M. K. *TensorBoard.dev*. 2022. Disponível em: <https://tensorboard.dev/experiment/2tgfgF1sRgyAxaMMEfMwDA>. Acesso em: 25/06/2022. Citado na página 39.

CARISSIMI, N.; BEYAN, C.; MURINO, V. A multi-view learning approach to deception detection. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. [S.l.: s.n.], 2018. p. 599–606. Citado 4 vezes nas páginas 23, 24, 33 e 38.

CARREIRA, J. et al. *A Short Note about Kinetics-600*. 2018. Acesso em: 25/06/2022. Disponível em: <https://arxiv.org/abs/1808.01340>. Citado na página 40.

CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 20, n. 3, p. 273–297, sep 1995. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022627411411>. Citado na página 19.

DELGADO, A. C. et al. Deception detection using machine learning. In: *Proceedings of the 54th Hawaii International Conference on System Sciences*. Hawaii: [s.n.], 2021. Citado na página 19.

DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado na página 25.

DEPAULO, B. et al. Lying in everyday life. *Journal of personality and social psychology*, v. 70, p. 979–95, 06 1996. Citado na página 16.

DING, M. et al. Face-focused cross-stream network for deception detection in videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 7794–7803, 2019. Citado 6 vezes nas páginas 22, 23, 33, 34, 38 e 46.

DODIA, S. et al. Lie detection using extreme learning machine: A concealed information test based on short-time fourier transform and binary bat optimization using a novel fitness function. *Computational Intelligence*, v. 36, n. 2, p. 637–658, 2020. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12256>. Citado na página 20.

EKMAN, P.; O'SULLIVAN, M. Who can catch a liar? *The American psychologist*, v. 46 9, p. 913–20, 1991. Citado na página 16.

EXPO, M. *Capacete para EEG 256 canais*. 2022. Disponível em: <https://www.medicalexpo.com/pt/prod/compumedics-neuroscan/product-79144-501556.html>. Acesso em: 25/06/2022. Citado na página 21.

FBI. *FBI Media*. 2022. Disponível em: <https://multimedia.fbi.gov/>. Acesso em: 25/06/2022. Citado na página 18.

FEICHTENHOFER, C. et al. SlowFast networks for video recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. Citado 5 vezes nas páginas 29, 30, 31, 38 e 41.

FIEDLER, K.; SCHMID, J.; STAHL, T. What is the current truth about polygraph lie detection? *Basic and Applied Social Psychology*, Routledge, v. 24, n. 4, p. 313–324, 2002. Disponível em: <https://doi.org/10.1207/S15324834BASP2404_6>. Citado 3 vezes nas páginas 13, 17 e 18.

FITZPATRICK, E.; BACHENKO, J. Building a data collection for deception research. In: *Proceedings of the EACL 2012 Workshop on Computational Approaches to Deception Detection*. Avignon, France: Association for Computational Linguistics, 2012. p. 31–38. Disponível em: <https://aclanthology.org/W12-0405.pdf>. Citado na página 25.

FOUNDATION, T. A. S. *Apache MXNet*. 2022. Acesso em: 25/06/2022. Disponível em: <https://mxnet.apache.org/versions/1.9.1/>. Citado na página 32.

GAO, J. et al. A novel algorithm to enhance p300 in single trials: Application to lie detection using F-Score and SVM. *PLOS ONE*, Public Library of Science, v. 9, n. 11, p. 1–15, 11 2014. Disponível em: <https://doi.org/10.1371/journal.pone.0109700>. Citado na página 20.

GOGATE, M.; ADEEL, A.; HUSSAIN, A. Deep learning driven multimodal fusion for automated deception detection. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2017. p. 1–6. Citado 2 vezes nas páginas 23 e 24.

GOKHMAN, S. et al. In search of a gold standard in studies of deception. In: *Proceedings of the EACL 2012 Workshop on Computational Approaches to Deception Detection*. Avignon, France: Association for Computational Linguistics, 2012. p. 23–30. Disponível em: <https://aclanthology.org/W12-0405.pdf>. Citado na página 25.

GONZALEZ-BILLANDON, J. et al. Can a robot catch you lying? a machine learning system to detect lies during interactions. *Frontiers in Robotics and AI*, v. 6, 2019. ISSN 2296-9144. Disponível em: <https://www.frontiersin.org/article/10.3389/frobt.2019.00064>. Citado 2 vezes nas páginas 20 e 21.

GOODFELLOW, I. J. et al. Generative adversarial nets. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 2672–2680. Citado na página 22.

GROSS, S.; SHAFFER, M. Exonerations in the United States, 1989–2012. *SSRN Electronic Journal*, 06 2012. Citado 2 vezes nas páginas 16 e 26.

GUO, J. et al. GluonCV and GluonNLP: Deep learning in computer vision and natural language processing. *Journal of Machine Learning Research*, v. 21, n. 23, p. 1–7, 2020. Disponível em: <http://jmlr.org/papers/v21/19-429.html>. Citado na página 32.

HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 22.

HERATH, S.; HARANDI, M.; PORIKLI, F. Going deeper into action recognition: A survey. *Image and Vision Computing*, v. 60, p. 4–21, 2017. ISSN 0262-8856. Regularization Techniques for High-Dimensional Data Analysis. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0262885617300343>. Citado na página 28.

HERNÁNDEZ-CASTAÑEDA, Á. et al. Cross-domain deception detection using support vector networks. *Soft Comput.*, Springer-Verlag, Berlin, Heidelberg, v. 21, n. 3, p. 585–595, feb 2017. ISSN 1432-7643. Disponível em: <https://doi.org/10.1007/s00500-016-2409-2>. Citado na página 19.

HERNANDEZ-REYNOSO, A. G.; GARCIA-GONZALEZ, A. Deception detection using fNIR imaging and neural networks. *Reporte Tecnico RT-0003-2013*, Junio 2013. Citado na página 20.

HO, S. M.; HANCOCK, J. T. Context in a bottle: Language-action cues in spontaneous computer-mediated deception. *Computers in Human Behavior*, v. 91, p. 33–41, 2019. ISSN 0747-5632. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0747563218304485>. Citado na página 19.

HONTS, C.; RASKIN, D.; KIRCHER, J. Mental and physical countermeasures reduce the accuracy of polygraph tests. *The Journal of applied psychology*, v. 79, p. 252–9, 05 1994. Citado na página 18.

JAISWAL, M.; TABIBU, S.; BAJPAI, R. The truth and nothing but the truth: Multimodal analysis for deception detection. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. [S.l.: s.n.], 2016. p. 938–943. Citado 4 vezes nas páginas 22, 23, 24 e 33.

JI, S. et al. 3d convolutional neural networks for human action recognition. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010. (ICML'10), p. 495–502. ISBN 9781605589077. Citado na página 29.

JIANG, B. et al. STM: Spatiotemporal and motion encoding for action recognition. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. p. 2000–2009. Citado na página 30.

KARPATHY, A. et al. Large-scale video classification with convolutional neural networks. In: *CVPR*. [S.l.: s.n.], 2014. Citado na página 25.

KAY, W. et al. *The Kinetics Human Action Video Dataset*. 2017. Acesso em: 25/06/2022. Disponível em: <https://arxiv.org/abs/1705.06950>. Citado 5 vezes nas páginas 28, 38, 40, 50 e 67.

KDENLIVE. *Kdenlive*. 2022. Acesso em: 25/06/2022. Disponível em: <https://kdenlive.org/en/>. Citado 2 vezes nas páginas 32 e 62.

KHAN, W. et al. Deception in the eyes of deceiver: A computer vision and machine learning based automated deception detection. *Expert Systems with Applications*, v. 169, p. 114341, 2021. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417420310289>. Citado 5 vezes nas páginas 13, 16, 17, 22 e 25.

KHURANA, R.; KUSHWAHA, A. K. S. Deep learning approaches for human activity recognition in video surveillance - a survey. In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. [S.l.: s.n.], 2018. p. 542–544. Citado na página 28.

KLEINBERG, B.; VERSCHUERE, B. How humans impair automated deception detection performance. *Acta Psychologica*, v. 213, p. 103250, 02 2021. Citado na página 26.

KORSTANJE, J. *The F1 score*. 2021. Disponível em: <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>. Acesso em: 25/06/2022. Citado na página 20.

KRISHNAMURTHY, G. et al. *A Deep Learning Approach for Multimodal Deception Detection*. 2018. Acesso em: 25/06/2022. Disponível em: <http://arxiv.org/abs/1803.00344>. Citado 6 vezes nas páginas 22, 23, 24, 33, 34 e 46.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 60, n. 6, p. 84–90, may 2017. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3065386>. Citado na página 24.

KWON, H. et al. MotionSqueeze: Neural motion feature learning for video understanding. In: *ECCV*. [S.l.: s.n.], 2020. Citado 2 vezes nas páginas 28 e 30.

LAI, V.; TAN, C. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2019. (FAT* '19), p. 29–38. ISBN 9781450361255. Disponível em: <https://doi.org/10.1145/3287560.3287590>. Citado 2 vezes nas páginas 19 e 26.

LI, Y. et al. TEA: Temporal excitation and aggregation for action recognition. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020. p. 906–915. Citado na página 30.

LIN, J.; GAN, C.; HAN, S. TSM: Temporal shift module for efficient video understanding. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. p. 7082–7092. Citado 2 vezes nas páginas 29 e 30.

LIU, Z. et al. TEInet: Towards an efficient architecture for video recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 34, p. 11669–11676, 04 2020. Citado na página 30.

MANN, S.; VRIJ, A.; BULL, R. Detecting true lies: Police officers' ability to detect suspects' lies. *The Journal of applied psychology*, v. 89, p. 137–49, 03 2004. Citado 4 vezes nas páginas 15, 16, 17 e 25.

MATHUR, L.; MATARIC, M. J. *Introducing Representations of Facial Affect in Automated Multimodal Deception Detection*. 2020. Acesso em: 25/06/2022. Disponível em: <http://arxiv.org/abs/2008.13369>. Citado 4 vezes nas páginas 22, 23, 24 e 33.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfan van der; MILLMAN Jarrod (Ed.). *Proceedings of the 9th Python in Science Conference*. [S.l.: s.n.], 2010. p. 56 – 61. Citado na página 32.

MENDELS, G. et al. Hybrid Acoustic-Lexical Deep Learning Approach for Deception Detection. In: *Proc. Interspeech 2017*. Stockholm, Sweden: [s.n.], 2017. p. 1472–1476. Citado 2 vezes nas páginas 20 e 21.

MICROSOFT. *Visual Studio Code*. 2022. Acesso em: 25/06/2022. Disponível em: <https://code.visualstudio.com/>. Citado na página 33.

MIHALCEA, R. *Rada Mihalcea: Downloads*. 2016. Acesso em: 25/06/2022. Disponível em: <https://web.eecs.umich.edu/~mihalcea/downloads.html>. Citado na página 33.

NARKHEDE, S. *Understanding AUC - ROC Curve*. 2018. Disponível em: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>. Acesso em: 25/06/2022. Citado na página 21.

NG, J. Y.-H. et al. Beyond short snippets: Deep networks for video classification. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 4694–4702. Citado na página 28.

NGÔ, M. et al. *Deception Detection by 2D-to-3D Face Reconstruction from Videos*. 2018. Acesso em: 25/06/2022. Disponível em: <http://arxiv.org/abs/1812.10558>. Citado 5 vezes nas páginas 15, 23, 24, 33 e 38.

OSWALD, M. Technologies in the twilight zone: early lie detectors, machine learning and reformist legal realism. *International Review of Law, Computers & Technology*, Routledge, v. 34, n. 2, p. 214–231, 2020. Disponível em: <https://doi.org/10.1080/13600869.2020.1733758>. Citado na página 17.

PASZKE, A. et al. PyTorch: An imperative style, high-performance deep learning library. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>. Citado na página 52.

PÉREZ-ROSAS, V. et al. Deception detection using real-life trial data. In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. New York, NY, USA: Association for Computing Machinery, 2015. (ICMI '15), p. 59–66. ISBN 9781450339124. Disponível em: <https://doi.org/10.1145/2818346.2820758>. Citado 2 vezes nas páginas 22 e 23.

REN, Q. et al. A survey on video classification methods based on deep learning. *DEStech Transactions on Computer Science and Engineering*, 12 2019. Citado na página 28.

RUITER, B. de; KACHERGIS, G. *The Mafiascum Dataset: A Large Text Corpus for Deception Detection*. 2018. Acesso em: 25/06/2022. Disponível em: <http://arxiv.org/abs/1811.07851>. Citado na página 19.

RUSSELL, J. A circumplex model of affect. *Journal of Personality and Social Psychology*, v. 39, p. 1161–1178, 12 1980. Citado na página 24.

SHARMA, S. *Deep Learning Architectures for Action Recognition*. 2020. Acesso em: 25/06/2022. Disponível em: <https://towardsdatascience.com/deep-learning-architectures-for-action-recognition-83e5061ddf90>. Citado na página 28.

SIGURDSSON, G. A. et al. Hollywood in homes: Crowdsourcing data collection for activity understanding. In: LEIBE, B. et al. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 510–526. ISBN 978-3-319-46448-0. Citado na página 40.

SIMONYAN, K.; ZISSERMAN, A. Two-stream convolutional networks for action recognition in videos. In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. v. 27. Disponível em: <https://proceedings.neurips.cc/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf>. Citado na página 28.

SOLDNER, F.; PÉREZ-ROSAS, V.; MIHALCEA, R. Box of lies: Multimodal deception detection in dialogues. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 1768–1777. Disponível em: <https://aclanthology.org/N19-1175>. Citado na página 22.

TRAN, D. et al. *Video Classification with Channel-Separated Convolutional Networks*. 2019. Acesso em: 25/06/2022. Disponível em: <http://arxiv.org/abs/1904.02811>. Citado 2 vezes nas páginas 29 e 30.

VENKATESH, S. et al. Video based deception detection using deep recurrent convolutional neural network. In: NAIN, N.; VIPPARTHI, S. K.; RAMAN, B. (Ed.). *Computer Vision and Image Processing*. Singapore: Springer Singapore, 2020. p. 163–169. ISBN 978-981-15-4018-9. Citado 2 vezes nas páginas 22 e 23.

VRIJ, A. *Detecting Lies and Deceit: Pitfalls and Opportunities*. Wiley, 2008. (Wiley Series in Psychology of Crime, Policing and Law). ISBN 9780470516256. Disponível em: <https://books.google.com.br/books?id=20pg76wmAucC>. Citado 7 vezes nas páginas 13, 15, 16, 17, 22, 24 e 25.

WANG, H. et al. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, Springer Verlag, v. 119, n. 3, p. 219–238, set. 2016. Disponível em: <https://hal.inria.fr/hal-01145834>. Citado na página 24.

WANG, X. et al. Non-local neural networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 7794–7803. Citado na página 31.

WU, Z. et al. Deception detection in videos. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. [S.l.]: AAAI Press, 2018. (AAAI'18/IAAI'18/EAAI'18). ISBN 978-1-57735-800-8. Citado 7 vezes nas páginas 22, 23, 24, 28, 33, 34 e 46.

XIAO, X.; XU, D.; WAN, W. Overview: Video recognition from handcrafted method to deep learning method. In: *2016 International Conference on Audio, Language and Image Processing (ICALIP)*. [S.l.: s.n.], 2016. p. 646–651. Citado na página 28.

XU, C.; TAO, D.; XU, C. *A Survey on Multi-view Learning*. 2013. Acesso em: 25/06/2022. Disponível em: <https://arxiv.org/abs/1304.5634>. Citado na página 24.

XUE, A.; ROHDE, H.; FINKELSTEIN, A. *An Acoustic Automated Lie Detector*. 2019. Acesso em: 25/06/2022. Disponível em: <https://www.cs.princeton.edu/sites/default/files/alice_xue_spring_2019.pdf>. Citado 2 vezes nas páginas 20 e 21.

ZHANG, Z. et al. Deep learning based human action recognition: A survey. In: *2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 3780–3785. Citado na página 28.

ZHU, Y. et al. Hidden two-stream convolutional networks for action recognition. In: ____. *Computer Vision – ACCV 2018*. Cham: Springer International Publishing, 2019. p. 363–378. ISBN 978-3-030-20893-6. Citado 2 vezes nas páginas 28 e 30.

ZHU, Y. et al. *A Comprehensive Study of Deep Video Action Recognition*. 2020. Acesso em: 25/06/2022. Disponível em: <https://arxiv.org/abs/2012.06567>. Citado 3 vezes nas páginas 28, 29 e 30.

Apêndices

APÊNDICE A – Modificações no RLT

O RLT conta originalmente com 121 vídeos, dos quais 61 apresentam mentiras e 60 apresentam verdades. Os vídeos que apresentam mentiras são nomeados no padrão “trial_lie_0XX”, com XX variando de 01 a 61, e os vídeos que apresentam verdades são nomeados no padrão “trial_truth_0XX”, com XX variando de 01 a 60.

Tendo isso em vista, os seguintes vídeos foram descartados do *dataset* por serem inadequados à tarefa de classificação (a qual requer que apenas a face do indivíduo que efetua o testemunho mentiroso ou verdadeiro esteja visível):

- trial_lie_035: apresenta cerca de 6 indivíduos;
- trial_lie_045: apresenta 3 indivíduos em primeiro plano;
- trial_lie_050: vídeo extraído de um programa televisivo no qual o indivíduo mentiroso está sendo interrogado. Uma grande caixa de texto dificulta a visualização do indivíduo e a parte traseira da cabeça do interrogador oculta a face do indivíduo durante uma parcela significativa do vídeo;
- trial_lie_052: vídeo extraído de um programa televisivo com grande número de cortes, utilização de diferentes ângulos e exibição da face da apresentadora do programa televisivo;
- trial_lie_053: vídeo extraído de um canal do YouTube com elementos visuais se sobrepondo a um vídeo obtido a partir de um programa televisivo. Vídeo conta com muitos cortes, elementos visuais que podem ser considerados ruído, além de apresentar o indivíduo mentiroso por um intervalo de tempo curto e com a face pouco visível;
- trial_lie_055: vídeo apresenta 2 indivíduos em primeiro plano;
- trial_lie_056: vídeo apresenta 2 indivíduos em primeiro plano;
- trial_lie_060: vídeo apresenta 4 indivíduos, 2 dos quais em primeiro plano em gravação ao ar livre;
- trial_truth_026: vídeo apresenta 4 indivíduos, com 2 em primeiro plano;
- trial_truth_029: face do indivíduo fora de foco durante a maior parte do vídeo;
- trial_truth_041: face do indivíduo fora de foco durante a maior parte do vídeo.

Após a remoção de tais vídeos, o *dataset* passou a contar com 110 vídeos, dos quais 57 apresentam verdades e 53 apresentam mentiras. Dentre tais vídeos, alguns contavam com

pequenos trechos considerados inadequados à tarefa de classificação, seja por apresentarem outros indivíduos, por não focarem no indivíduo relatando a verdade ou mentira, ou por apresentarem ruídos visuais. Assim, os seguintes vídeos tiveram as respectivas edições realizadas no editor Kdenlive (KDENLIVE, 2022):

- trial_lie_005: *frames* com ruído visual eliminados do segundo 27;
- trial_lie_010: primeiros 3 segundos removidos;
- trial_lie_017: remoção do segundo 19 ao segundo 30;
- trial_lie_021: *frames* iniciais e finais do vídeo removidos;
- trial_lie_022: *frames* finais removidos;
- trial_lie_023: *frames* iniciais removidos;
- trial_lie_037: *frames* com ruído visual eliminados do segundo 20;
- trial_lie_041: *frames* iniciais e finais do vídeo removidos;
- trial_lie_042: *frames* finais removidos;
- trial_lie_043: corte realizado a partir do segundo 7 do vídeo;
- trial_lie_044: *frames* iniciais e finais do vídeo removidos;
- trial_lie_046: primeiros 6 segundos removidos;
- trial_lie_047: *frames* finais removidos;
- trial_lie_051: *frames* iniciais e finais do vídeo removidos;
- trial_lie_054: corte realizado a partir do segundo 10 do vídeo;
- trial_lie_057: *frames* iniciais removidos;
- trial_lie_058: *frames* finais removidos;
- trial_lie_061: *frames* iniciais e finais do vídeo removidos;
- trial_truth_001: *frames* finais removidos;
- trial_truth_006: *frames* iniciais removidos;
- trial_truth_007: corte do segundo 9 ao segundo 12 e do segundo 44 ao segundo 46;
- trial_truth_008: *frames* iniciais e finais do vídeo removidos;
- trial_truth_009: *frames* iniciais removidos;

- trial_truth_042: primeiros 9 segundos do vídeo removidos;
- trial_truth_051: *frames* iniciais removidos.

APÊNDICE B – Relação de Vídeos por Indivíduo

O Quadro 4 apresenta a relação entre os diferentes indivíduos (identificados a partir de números de identificação) e os vídeos nos quais eles são retratados considerando o *dataset* RLT após a etapa de seleção de vídeos.

Quadro 4 – Relação entre os diferentes indivíduos e os vídeos que os retratam no *dataset* RLT após a etapa de seleção de vídeos.

Indivíduo	**Arquivo**
1	trial_lie_001
1	trial_lie_002
1	trial_lie_003
1	trial_lie_004
1	trial_lie_005
1	trial_lie_006
1	trial_truth_057
1	trial_truth_058
1	trial_truth_059
2	trial_lie_007
2	trial_lie_008
2	trial_lie_009
2	trial_lie_010
2	trial_lie_011
2	trial_lie_012
2	trial_lie_013
2	trial_truth_003
2	trial_truth_004
2	trial_truth_005
2	trial_truth_006
2	trial_truth_007
3	trial_lie_014
3	trial_lie_015
3	trial_lie_016
3	trial_lie_017
3	trial_lie_018
3	trial_lie_019
3	trial_lie_020
3	trial_lie_021
3	trial_lie_022
3	trial_lie_023
3	trial_lie_024
3	trial_lie_025
3	trial_lie_026
3	trial_lie_027
3	trial_lie_028
3	trial_lie_029
3	trial_lie_030
3	trial_lie_031
3	trial_truth_054
3	trial_truth_055
3	trial_truth_056
4	trial_lie_032
5	trial_lie_033
6	trial_lie_034
7	trial_lie_036
7	trial_lie_037
7	trial_lie_038
7	trial_lie_039
7	trial_lie_040
7	trial_truth_053
7	trial_truth_060
8	trial_lie_041
9	trial_lie_042
10	trial_lie_043
11	trial_lie_044
12	trial_lie_046
12	trial_lie_047
12	trial_lie_048
12	trial_lie_049
13	trial_lie_051
14	trial_lie_054
15	trial_lie_057
16	trial_lie_058
17	trial_lie_059
18	trial_truth_049
19	trial_lie_061
20	trial_truth_001
21	trial_truth_002
22	trial_truth_008
22	trial_truth_009
22	trial_truth_010
22	trial_truth_011
22	trial_truth_012
22	trial_truth_013
22	trial_truth_014
22	trial_truth_015
23	trial_truth_016
24	trial_truth_017
25	trial_truth_018
25	trial_truth_019
26	trial_truth_020
27	trial_truth_021
28	trial_truth_022
29	trial_truth_023
30	trial_truth_024
30	trial_truth_025
31	trial_truth_027
32	trial_truth_028
33	trial_truth_031
34	trial_truth_030
35	trial_truth_032
36	trial_truth_033
36	trial_truth_034
37	trial_truth_035
37	trial_truth_036
38	trial_truth_037
39	trial_truth_038
40	trial_truth_039
41	trial_truth_040
42	trial_truth_042
43	trial_truth_043
44	trial_truth_044
45	trial_truth_045
46	trial_truth_046
47	trial_truth_047
48	trial_truth_048
49	trial_truth_050
50	trial_truth_051
51	trial_truth_052

APÊNDICE C – Relação de Indivíduos por Conjunto

Os seguintes indivíduos foram alocados para cada conjunto que compõe a separação A:

- Treino: 1, 3, 5, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 21, 22, 23, 24, 26, 27, 29, 30, 31, 33, 34, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, 51;
- Validação: 2, 6, 20, 28, 47;
- Teste: 4, 7, 10, 16, 25, 32, 35, 36.

Já para a separação B, os seguintes indivíduos foram alocados para cada *fold*:

- *Fold* 0: 3, 20;
- *Fold* 1: 7, 11, 12, 19, 25, 30, 33, 36, 38, 48;
- *Fold* 2: 2, 4, 6, 9, 16, 37, 41, 42, 46, 49;
- *Fold* 3: 1, 5, 8, 10, 15, 17, 22;
- *Fold* 4: 13, 14, 18, 21, 23, 24, 26, 27, 28, 29, 31, 32, 34, 35, 39, 40, 43, 44, 45, 47, 50, 51.

Por sua vez, a separação C, que não efetuou a divisão dos indivíduos por *fold*, teve vídeos dos seguintes indivíduos alocados para cada *fold*:

- *Fold* 0: 1, 2, 3, 7, 8, 15, 19, 20, 21, 22, 25, 28, 36, 48, 51;
- *Fold* 1: 1, 2, 3, 4, 7, 11, 12, 16, 22, 24, 25, 31, 35, 37, 43, 50;
- *Fold* 2: 1, 2, 3, 7, 9, 10, 12, 14, 22, 30, 34, 37, 42, 44, 45, 49;
- *Fold* 3: 1, 2, 3, 5, 6, 7, 12, 13, 18, 22, 27, 29, 30, 33, 39;
- *Fold* 4: 1, 2, 3, 7, 12, 17, 22, 23, 26, 32, 36, 38, 40, 41, 46, 47.

APÊNDICE D – Resultados da Busca de Hiperparâmetros

Os Quadros 5 e 6 apresentam as acurácia de treino e validação obtidas a partir dos diferentes hiperparâmetros avaliados. Em ambos, a estratégia de decaimento da taxa de aprendizado e a arquitetura SlowFast seguem a seguinte representação:

- A: Sem decaimento;
- B: Decaimento a cada 10 épocas até a época 100;
- C: Decaimento a cada 10 épocas até a época 200;
- D: Decaimento nas épocas 40, 80 e 100;
- 4x16: 4x16 ResNet50 sem pré-treino;
- 4x16K: 4x16 ResNet50 com pré-treino no Kinetics-400 (KAY et al., 2017);
- 8x8K: 8x8 ResNet50 com pré-treino no Kinetics-400.

Quadro 5 – Resultados da Busca de Hiperparâmetros - Parte 1. α - Taxa de Aprendizado, Estrat. - Estratégia de Decaimento da Taxa de Aprendizado, Otim. - Otimização, Val. - Validação. Os valores de Estrat. e SlowFast são explicados no texto.

ID	α	**Estrat.**	**Otim.**	**Weight Decay**	**SlowFast**	**Épocas**	**Treino**	**Val.**
0	0,001	D	Adam	0,0001	4x16K	100	0,9615	0,5206
1	0,001	D	SGD	0,001	4x16K	100	0,8974	0,5450
2	0,001	D	SGD	0,01	4x16K	100	0,8718	0,5250
3	0,001	D	SGD	0,1	4x16K	100	0,7051	0,5313
4	0,005	A	SGD	0,0001	4x16K	100	0,7821	0,5600
5	0,005	B	SGD	0,0001	4x16K	100	0,6667	0,5356
6	0,005	D	SGD	0,0001	4x16K	100	0,7821	0,5988
7	0,001	A	SGD	0,0001	4x16K	100	0,8974	0,4544
8	0,001	B	SGD	0,0001	4x16K	100	0,6154	0,5913
9	0,001	D	SGD	0,0001	4x16K	100	0,8333	0,4319
10	0,0005	A	SGD	0,0001	4x16K	100	0,8590	0,4488
11	0,0005	B	SGD	0,0001	4x16K	100	0,5897	0,6038
12	0,0005	D	SGD	0,0001	4x16K	100	0,7949	0,4788
13	0,0001	A	SGD	0,0001	4x16K	100	0,7179	0,4994
14	0,0001	B	SGD	0,0001	4x16K	100	0,5641	0,5025
15	0,0001	D	SGD	0,0001	4x16K	100	0,5641	0,5544
16	0,001	B	SGD	0,1	4x16K	100	0,7564	0,3994
17	0,001	B	SGD	0,01	4x16K	100	0,7308	0,5825
18	0,001	B	SGD	0,001	4x16K	100	0,7308	0,4338
19	0,0005	B	SGD	0,1	4x16K	100	0,7051	0,5025
20	0,0005	B	SGD	0,01	4x16K	100	0,6282	0,4856
21	0,0005	B	SGD	0,001	4x16K	100	0,5769	0,4425
22	0,005	D	SGD	0,1	4x16K	100	0,5256	0,4975
23	0,005	D	SGD	0,01	4x16K	100	0,9231	0,5463
24	0,005	D	SGD	0,001	4x16K	100	0,7692	0,5138
25	0,005	D	SGD	0,0001	4x16	100	0,9872	0,5181
26	0,001	B	SGD	0,0001	4x16	100	0,9872	0,5050
27	0,0005	B	SGD	0,0001	4x16	100	0,9231	0,4956
28	0,001	B	SGD	0,0001	8x8K	100	0,8077	0,5394
29	0,0005	B	SGD	0,0001	8x8K	100	0,6154	0,4588
30	0,005	D	SGD	0,0001	8x8K	100	0,7564	0,5781
31	0,005	B	Adam	0,001	4x16K	100	0,6026	0,5656
32	0,005	B	Adam	0,0001	4x16K	100	0,6410	0,5794
33	0,001	B	Adam	0,001	4x16K	100	0,8590	0,5813
34	0,001	B	Adam	0,0001	4x16K	100	0,7308	0,5188

Quadro 6 – Resultados da Busca de Hiperparâmetros - Parte 2. α - Taxa de Aprendizado, Estrat. - Estratégia de Decaimento da Taxa de Aprendizado, Otim. - Otimização, Val. - Validação. Os valores de Estrat. e SlowFast são explicados no texto.

ID	α	**Estrat.**	**Otim.**	**Weight Decay**	**SlowFast**	**Épocas**	**Treino**	**Val.**
35	0,0005	B	Adam	0,001	4x16K	100	0,7692	0,3975
36	0,0005	B	Adam	0,0001	4x16K	100	0,8333	0,4413
37	0,0001	B	Adam	0,001	4x16K	100	0,7949	0,4394
38	0,0001	B	Adam	0,0001	4x16K	100	0,8462	0,4681
39	0,005	B	RMSProp	0,001	4x16K	100	0,6026	0,4250
40	0,005	B	RMSProp	0,0001	4x16K	100	0,6795	0,5213
41	0,001	B	RMSProp	0,001	4x16K	100	0,7821	0,5356
42	0,001	B	RMSProp	0,0001	4x16K	100	0,8077	0,5556
43	0,0005	B	RMSProp	0,001	4x16K	100	0,8846	0,4294
44	0,0005	B	RMSProp	0,0001	4x16K	100	0,8590	0,5613
45	0,0001	B	RMSProp	0,001	4x16K	100	0,7949	0,4444
46	0,0001	B	RMSProp	0,0001	4x16K	100	0,8333	0,4681
47	0,005	D	Adam	0,001	4x16K	100	0,8718	0,4694
48	0,005	D	Adam	0,0001	4x16K	100	0,6795	0,5350
49	0,001	D	Adam	0,001	4x16K	100	0,9231	0,4906
50	0,001	D	Adam	0,0001	4x16K	100	0,9231	0,5381
51	0,0005	D	Adam	0,001	4x16K	100	0,9487	0,4694
52	0,0005	D	Adam	0,0001	4x16K	100	0,9744	0,3950
53	0,0001	D	Adam	0,001	4x16K	100	0,9872	0,4375
54	0,0001	D	Adam	0,0001	4x16K	100	0,9872	0,5144
55	0,005	D	RMSProp	0,001	4x16K	100	0,6795	0,4669
56	0,005	D	RMSProp	0,0001	4x16K	100	0,7436	0,5069
57	0,001	D	RMSProp	0,001	4x16K	100	0,9487	0,5519
58	0,001	D	RMSProp	0,0001	4x16K	100	0,9487	0,5638
59	0,0005	D	RMSProp	0,001	4x16K	100	0,9744	0,4325
60	0,0005	D	RMSProp	0,0001	4x16K	100	0,9872	0,4413
61	0,0001	D	RMSProp	0,001	4x16K	100	0,9744	0,4419
62	0,0001	D	RMSProp	0,0001	4x16K	100	0,9872	0,4644
63	0,001	C	SGD	0,0001	4x16K	200	0,6667	0,5088
64	0,0005	C	SGD	0,0001	4x16K	200	0,6795	0,5066
65	0,001	B	SGD	0,0001	4x16K	200	0,6923	0,6138
66	0,0005	B	SGD	0,0001	4x16K	200	0,6282	0,4644
67	0,005	D	SGD	0,0001	4x16K	200	0,7308	0,6041
68	0,005	D	SGD	0,0001	8x8K	200	0,7436	0,5484

APÊNDICE E – Matrizes de Confusão para Testes na Separação B

As Tabelas 16, 17, 18, 19 e 20 apresentam os resultados de execução das combinações de hiperparâmetros de IDs 6, 8, 11, 65 e 67, respectivamente, sobre as *folds* da separação B.

Tabela 16 – Matrizes de confusão para cada *fold* testada para a combinação de hiperparâmetros de ID 6 conforme separação B. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	0	18 (81,8%)	18
	V	2 (9,1%)	2 (9,1%)	4
	Total	2	20	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	4 (18,2%)	7 (31,8%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	7	15	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	9 (40,9%)	2 (9,1%)	11
	V	8 (36,4%)	3 (13,6%)	11
	Total	17	5	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	9 (40,9%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	5	17	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	1 (4,5%)	1 (4,5%)	2
	V	16 (72,7%)	4 (18,2%)	20
	Total	17	5	22

Tabela 17 – Matrizes de confusão para cada *fold* testada para a combinação de hiperparâmetros de ID 8 conforme separação B. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	18 (81,8%)	0	18
	V	2 (9,1%)	2 (9,1%)	4
	Total	20	2	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	9 (40,9%)	2 (9,1%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	16	6	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	4 (18,2%)	7 (31,8%)	11
	V	5 (22,7%)	6 (27,3%)	11
	Total	9	13	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	1 (4,5%)	10 (45,5%)	11
	V	4 (18,2%)	7 (31,8%)	11
	Total	5	17	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	0	2
	V	7 (31,8%)	13 (59,1%)	20
	Total	9	13	22

Tabela 18 – Matrizes de confusão para cada *fold* testada para a combinação de hiperparâmetros de ID 11 conforme separação B. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	16 (72,7%)	2 (9,1%)	18
	V	0	4 (18,2%)	4
	Total	16	6	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	8 (36,4%)	3 (13,6%)	11
	Total	15	7	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	10 (45,5%)	1 (4,5%)	11
	V	6 (27,3%)	5 (22,7%)	11
	Total	16	6	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	9 (40,9%)	11
	V	8 (36,4%)	3 (13,6%)	11
	Total	10	12	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	0	2
	V	7 (31,8%)	13 (59,1%)	20
	Total	9	13	22

Tabela 19 – Matrizes de confusão para cada *fold* testada para a combinação de hiperparâmetros de ID 65 conforme separação B. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	11 (50%)	18
	V	0	4 (18,2%)	4
	Total	7	15	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	5 (22,7%)	6 (27,3%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	8	14	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	9 (40,9%)	2 (9,1%)	11
	Total	16	6	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	1 (4,5%)	10 (45,5%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	8	14	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	1 (4,5%)	1 (4,5%)	2
	V	10 (45,5%)	10 (45,5%)	20
	Total	11	11	22

Tabela 20 – Matrizes de confusão para cada *fold* testada para a combinação de hiperparâmetros de ID 67 conforme separação B. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	1 (4,5%)	17 (77,3%)	18
	V	1 (4,5%)	3 (13,6%)	4
	Total	2	20	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	9 (40,9%)	11
	V	2 (9,1%)	9 (40,9%)	11
	Total	4	18	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	9 (40,9%)	2 (9,1%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	16	6	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	9 (40,9%)	11
	V	9 (40,9%)	2 (9,1%)	11
	Total	11	11	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	0	2 (9,1%)	2
	V	11 (50%)	9 (40,9%)	20
	Total	11	11	22

APÊNDICE F – Matrizes de Confusão para Testes na Separação C

As Tabelas 21, 22, 23, 24 e 25 apresentam os resultados de execução das combinações de hiperparâmetros de IDs 6, 8, 11, 65 e 67, respectivamente, sobre as *folds* da separação C.

Tabela 21 – Matrizes de confusão para cada *fold* balanceada testada para a combinação de hiperparâmetros de ID 6 conforme separação C. M indica Mentira e V indica Verdade.

FOLD 0

		Saída		
		M	V	Total
Classificação Real	M	8 (36,4%)	3 (13,6%)	11
	V	8 (36,4%)	3 (13,6%)	11
	Total	16	6	22

FOLD 1

		Saída		
		M	V	Total
Classificação Real	M	5 (22,7%)	6 (27,3%)	11
	V	4 (18,2%)	7 (31,8%)	11
	Total	9	13	22

FOLD 2

		Saída		
		M	V	Total
Classificação Real	M	8 (36,4%)	3 (13,6%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	11	11	22

FOLD 3

		Saída		
		M	V	Total
Classificação Real	M	9 (40,9%)	2 (9,1%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	16	6	22

FOLD 4

		Saída		
		M	V	Total
Classificação Real	M	5 (22,7%)	4 (18,2%)	9
	V	5 (22,7%)	8 (36,4%)	13
	Total	10	12	22

Tabela 22 – Matrizes de confusão para cada *fold* balanceada testada para a combinação de hiperparâmetros de ID 8 conforme separação C. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	6 (27,3%)	5 (22,7%)	11
	V	4 (18,2%)	7 (31,8%)	11
	Total	10	12	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	6 (27,3%)	5 (22,7%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	9	13	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	2 (9,1%)	9 (40,9%)	11
	Total	9	13	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	6 (27,3%)	5 (22,7%)	11
	Total	13	9	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	8 (36,4%)	1 (4,5%)	9
	V	3 (13,6%)	10 (45,5%)	13
	Total	11	11	22

Tabela 23 – Matrizes de confusão para cada *fold* balanceada testada para a combinação de hiperparâmetros de ID 11 conforme separação C. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	8 (36,4%)	3 (13,6%)	11
	V	6 (27,3%)	5 (22,7%)	11
	Total	14	8	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	8 (36,4%)	3 (13,6%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	15	7	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	14	8	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	10	12	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	5 (22,7%)	4 (18,2%)	9
	V	2 (9,1%)	11 (50%)	13
	Total	7	15	22

Tabela 24 – Matrizes de confusão para cada *fold* balanceada testada para a combinação de hiperparâmetros de ID 65 conforme separação C. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	8 (36,4%)	3 (13,6%)	11
	V	6 (27,3%)	5 (22,7%)	11
	Total	14	8	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	4 (18,2%)	7 (31,8%)	11
	V	6 (27,3%)	5 (22,7%)	11
	Total	10	12	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	3 (13,6%)	8 (36,4%)	11
	Total	10	12	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	6 (27,3%)	5 (22,7%)	11
	V	4 (18,2%)	7 (31,8%)	11
	Total	10	12	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	4 (18,2%)	5 (22,7%)	9
	V	1 (4,5%)	12 (54,5%)	13
	Total	5	17	22

Tabela 25 – Matrizes de confusão para cada *fold* balanceada testada para a combinação de hiperparâmetros de ID 67 conforme separação C. M indica Mentira e V indica Verdade.

FOLD 0

		Saída M	**Saída** V	Total
Classificação Real	M	6 (27,3%)	5 (22,7%)	11
	V	9 (40,9%)	2 (9,1%)	11
	Total	15	7	22

FOLD 1

		Saída M	**Saída** V	Total
Classificação Real	M	7 (31,8%)	4 (18,2%)	11
	V	7 (31,8%)	4 (18,2%)	11
	Total	14	8	22

FOLD 2

		Saída M	**Saída** V	Total
Classificação Real	M	6 (27,3%)	5 (22,7%)	11
	V	8 (36,4%)	3 (13,6%)	11
	Total	14	8	22

FOLD 3

		Saída M	**Saída** V	Total
Classificação Real	M	2 (9,1%)	9 (40,9%)	11
	V	2 (9,1%)	9 (40,9%)	11
	Total	4	18	22

FOLD 4

		Saída M	**Saída** V	Total
Classificação Real	M	5 (22,7%)	4 (18,2%)	9
	V	3 (13,6%)	10 (45,5%)	13
	Total	8	14	22