Universidade da Beira Interior
Departamento de Informática

Nº 130 - 2022: Avaliação de *Active Speaker Detection in Wild Conditions II*

Elaborado por:

João Pedro Roxo Salvado

Orientador:

Professor Doutor Hugo Proença

Co-Orientador:

Professor Doutor Tiago Roxo

8 de julho de 2022

Agradecimentos

Prestes a concluir uma grande etapa da minha vida, não poderia deixar passar este momento sem agradecer às pessoas que me ajudaram a crescer nestes últimos quatro anos.

Em primeiro lugar ao meu orientador Professor Doutor Hugo Pedro Martins Carriço Proença, do Departamento de Informática da Universidade da Beira Interior, quero agradecer pela oportunidade de participação neste projeto, bem como a disponibilidade prestada.

Ao meu co-orientador Professor Doutor Tiago Filipe Dias dos Santos Roxo, do Departamento de Informática da Universidade da Beira Interior, quero agradecer por toda a disponibilidade, ajuda e orientação prestada durante todo o projeto, permitindo o meu desenvolvimento intelectual.

Aos meus pais, quero agradeçer por toda a sua dedicação, ajuda e esforço, não só para garantir a minha educação nestes últimos quatro anos, como todo o meu desenvolvimento nos meus vinte e um anos de vida.

Aos meus amigos mais chegados, queria agradecer por todos os momentos que me ouviram, ajudaram e me fizeram rir.

Por fim gostaria de agradecer á pessoa mais importante sem a qual eu não seria o que sou hoje nem conseguiria chegar onde cheguei, um grande obrigado ao meu irmão Nuno Salvado por me dar forças, ajudar todos os dias e me incentivar a ser o melhor possível tanto a nível pessoal como profissional.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ASC *Active Speaker in Context*

ASD *Active Speaker Detection*

ASW *Active Speakers in the Wild*

AVA *Atomic Visual Action*

CSV *Comma-separated values*

CVAT *Computer Vision Annotation Tool*

FPS *Frames per Second*

LSTM *Long short-term memory*

mAP *Mean Average Precision*

MAAS *Multi-Modal Assignation for Active Speaker Detection*

SORT *Simple Object Realtime Tracking*

STE *Short-Term Encoder*

UBI Universidade da Beira Interior

WASD Wild Active Speaker Detection

Capítulo

1

Introdução

1.1 Enquadramento

Este projeto envolve o estudo e análise de um dos atuais modelos de *Active Speaker Detection* (ASD), o *Active Speaker in Context* (ASC), em ambiente não controlado. Em situação normal este modelo é aplicado em condições controladas e favoráveis. A elaboração deste projeto decorreu na unidade de projeto de final de curso da Licenciatura em Engenharia Informática na Universidade da Beira Interior (UBI).

1.2 Motivação UBI

Análise de comportamento humano é um tópico que tem sido alvo de muito interesse, dado à sua importância em perfilamento individual, previsão de ações, análise de contexto, entre outros aspetos. Tipicamente, a caracterização do comportamento humano pode ser segmentado na análise de diferentes fatores, nomeadamente, *soft biometrics*, e reconhecimento de objetos e ações. Um factor que nunca foi explorado em ambientes *wild*, ou seja, ambientes não controlados, são interações humanas, como por exemplo, perceber quem está a falar numa dada cena. Os atuais modelos estado de arte realizam *ASD* em ambientes controlados e favoráveis, não realizando uma testagem realista e intensiva deste. Para verificarmos as limitações destes modelos num ambiente *wild*, vamos utilizar vídeos gravitados para vídeo vigilância.

1.3 Objetivos

Este projeto tem como objetivo demonstrar as limitações dos atuais modelos de ASD através da criação de um *Wild Active Talking Speaker Detection dataset* que ao contrário de outros *datasets*, como o AVA-*ActiveSpeaker* em que os vídeos que o compõe são caracterizados por boa iluminação, bom acesso ás caras e boa qualidade de som, insere-se num cenário mais realista sendo composto por vídeos que nem sempre apresentam as bases necessárias para o modelo conseguir inferir da melhor maneira possível. Os objetivos definidos durante o desenvolvimento do projeto foram:

1. **Tarefa 1**: Análise de um dos modelos estado de arte para ASD e replicação dos resultados no *dataset* AVA-*ActiveSpeaker*;
2. **Tarefa 2**: Anotação do *dataset*;
3. **Tarefa 3**: Caracterização do *dataset* anotado em termos de diversos fatores que influenciam a *performance* do modelo estado de arte (oclusões, iluminação, estimativa de pose, entre outras);
4. **Tarefa 4**: Avaliação da *performance* do modelo estado de arte no *dataset* anotado;
5. **Tarefa 5**: Escrita do relatório de projeto.

1.4 Organização do Documento

Este documento encontra-se estruturado da seguinte forma:

1. **Introdução** – apresenta o projeto, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2. **Estado de Arte** – descreve algumas das implementações existentes. Estas são usadas para melhor perceber a forma como os atuais modelos funcionam.
3. **Implementação** – descreve todo o processo de implementação, como as ferramentas e processos utilizados durante a realização de anotações e conversão destas para o formato usado no *dataset* AVA-*ActiveSpeaker*.
4. **Avaliação do modelo *Active Speakers in Context*** – descreve o modelo estado de arte utilizado, os conjuntos de dados utilizados e como estes foram utilizados para avaliação do ASC, e os resultados obtidos desta avaliação e discussão sobre estes.

5. **Conclusões e Trabalho Futuro** – descrição as principais conclusões obtidas no final deste projeto e algumas sugestões para trabalho futuro.

Capítulo

2

Estado da Arte

2.1 Introdução

Neste capítulo abordamos alguns dos modelos e *datasets* atuais utilizados para ASD, de forma a analisarmos as abordagens e processos utilizados nos modelos e a lógica por trás das escolha de rotulamento dos vídeos que compõem cada conjunto de dados.

2.2 *Improved Active Speaker Detection based on Optical Flow*

Neste modelo é posto em prática o conceito de *Optical Flow*, sendo este a movimentação de objetos entre *frames* consecutivas, causado pelo movimento relativo entre o objeto e a câmara [1]. Mais especificamente é utilizado um dos dois tipos de *Optical Flow*, designado *Dense Optical Flow*.

Dense Optical Flow descreve os vetores de movimento de todos os píxeis de uma imagem e neste contexto será utilizado para extrair os movimentos faciais subtis.

Para o problema proposto, as entradas visuais e sonoras são processados de forma separada. Através de imagens faciais e *Dense Optical Flow*, são desenvolvidas duas estratégias de incorporação visual para fundir estes. As incorporações visuais e de áudio são concatenadas e fornecidas a uma rede de previsão, para fazer classificação binária das pessoas de estão ou não a falar. A eficácia deste modelo está diretamente dependente da qualidade das imagens fornecidas.

2.3 *AVA-ActiveSpeaker: An Audio-Visual Dataset for Active Speaker Detection*

ASD é um componente muito importante em algoritmos de análise de vídeo para diversas aplicações como *"speaker diarization"*, que consiste no rotulamento de gravações de vídeo ou áudio com classes que correspondem á identidade do locutor, sistemas de aprimoramento de fala, e interações humano-robô [2].

A falta de um *dataset* áudio-visual largo e rotulado cuidadosamente para as aplicações anteriormente mencionadas, dificultou a avaliação de algoritmos, o que tornou quaisquer comparações e melhorias extremamente difíceis. O *AVA-ActiveSpeaker* preenche esta necessidade, onde cada pessoa em cada *frame* é rotulada como estando a falar ou não e se o diálogo é audível.

2.4 *Look Who's Talking: Active Speaker Detection in the Wild*

Neste artigo é posto em causa a qualidade do *dataset AVA-ActiveSpeaker*, onde uma grande quantidade de vídeos que compõem este são dublados [3]. Como tal, há diversos casos onde o vídeo e som não estão sincronizados, o que leva a que o modelo aprenda a detetar se há fala no áudio reproduzido e se os lábios se estão a mexer, não detetando se os dois correspondem, ou seja, se ambos originam da mesma pessoa.

As secções de dublagem são marcadas como sendo "positivas", ou seja, a pessoa está a falar e no contexto de aprimoramento da relação áudio-vídeo, no reconhecimento da pessoa que está a falar e noutros casos, estas devem ser consideradas como "negativas". É proposto um novo *dataset Active Speakers in the Wild* (ASW) para ASD, que lida com as limitações do *AVA-ActiveSpeaker*.Os vídeos neste novo *dataset* são baseados no *VoxConverse*, um *dataset* com apenas registos de áudio, que resolve o problema de "quem fala quando".

2.5 *Is Someone Speaking? Exploring Long-term Temporal Features for Audio-visual Active Speaker Detection*

Um sistema de ASD depende de uma interpretação precisa de informação áudio e visual de longa e curta duração [4]. Ao contrário de outros sistemas que

tomam decisões instantaneamente usando vídeos de curta duração, é proposto uma *framework*, chamada *TalkNet*, que toma em consideração ambos os recursos de longo e curto termo. As experiências demonstram que o *TalkNet* atinge uma melhoria de 3.5% e 2.2% sobre sistemas como os *datasets AVA-ActiveSpeaker* e *Columbia ASD*, respetivamente.

Os sistemas existentes não beneficiaram de dois aspetos de informação disponíveis: as dinâmicas temporais do fluxo de vídeo e áudio, e a interação entre sinais de vídeo e áudio, que limita o escopo de aplicações, especialmente em cenários reais.

2.6 *VISUALVOICE: Audio-Visual Speech Separation with Cross-Modal Consistency*

Neste artigo é introduzida uma nova abordagem para extração de fala num ambiente áudio-visual, ou seja, extrair o discurso associado a uma pessoa mesmo havendo sons de fundo e/ou outras pessoas a falar [5].

Os outros métodos existentes focam-se em aprender o alinhamento entre os movimentos dos lábios do locutor e os sons que este gera, enquanto o método apresentado propõe aproveitar o rosto do locutor como uma variável adicional antes de isolar os sons mais prováveis de este produzir. O *VisualVoice* dá uso ao conceito de *cross-modal learning*, que diz que qualquer aprendizagem que advenha de uma modalidade sensorial pode ser melhorada com a informação de outros sentidos. O foco apenas nos movimentos dos lábios poderá induzir em erro quando estes se tornam falíveis (ex: a boca do locutor é tapada por um microfone). Para corrigir este problema, o *VisualVoice* procura "visualizar"a voz do locutor baseado na sua aparência, de forma a melhor isolar a voz desta.

2.7 *MAAS: Multi-modal Assignation for Active Speaker Detection*

Neste modelo de ASD é posto em prática o problema de atribuição, ou seja, o objetivo será combinar várias representações visuais de uma pessoa com apenas um único ficheiro de áudio e o conceito de aprendizagem multimodal, que envolve o ensino de um conceito usando mais do que um modo (visual, auditivo, leitura/escrita e cinestésico) [6]. A abordagem usada é a seguinte:

1. Detetar momentos de discurso numa pequena janela temporal.

2. Iterar sobre todos os locutores visíveis numa única *frame* e decidir qual deles será o mais provável de ser o locutor ativo dada a informação disponível.

3. Estender esta análise *frame* a *frame* juntamente com a dimensão temporal, aproveitando a consistência temporal de um vídeo, ou seja, numa janela de tempo a qualidade do vídeo mantém-se a mesma, para melhorar as previsões em cada *frame*.

2.8 Conclusões

O conteúdo deste capítulo descreve, como já foi mencionado, alguns dos diversos modelos e *datasets* usados em *ASD*, de forma a verificar que os atuais não diferem muito nas suas abordagens e escolhas. Os modelos dão especial foco ao alinhamento entre os movimentos dos lábios do locutor e os sons que este gera de forma a prever a pessoa que está a falar, já os *datasets* diferem ligeiramente na maneira como estes são rotulados.

Capítulo

3

Implementação

3.1 Introdução

Neste capítulo serão descritos todos os elementos envolvidos na implementação. É feita a descrição das ferramentas utilizadas para realização das anotações de *frames* de forma automatizada e descrição de todos os processos de anotação. Esta descrição foi dividida em vários pontos de interesse, sendo eles: o fluxo de anotações de *frames*, o fluxo de anotações de fala, conversão das anotações de *frames* para o formato utilizado pelo *dataset* AVA-*ActiveSpeaker*.

3.2 Ferramentas Utilizadas

3.2.1 YOLOv5

É uma família de modelos de deteção de objetos pré-treinados usando o COCO *dataset* em que divide imagens num sistema de rede [7]. Estes são capazes de deteção de objetos em tempo real com uma imensa precisão e muito mais. Cada célula nesta rede é responsável por detetar os objetos dentro de si mesma. O modelo pode ser treinado usando o nosso próprio *dataset*.

Em redes neuronais como o YOLO, para conseguirmos prever múltiplos objetos numa foto, a rede está de facto a fazer milhares de previsões e apenas mostra as que decidiu que são um objeto. Para fazer corretamente estas previsões são definidas na nossa imagem, centenas de caixas com diferentes tamanhos designadas de *anchor boxes*.

O YOLOv5 utiliza um algoritmo chamado *autoanchor* para gerar as *anchor boxes*, que recalcula estas para se ajustarem aos dados se as atuais não forem

favoráveis. Isto é usado em conjunto com o algoritmo *k-means*, sendo uma das razões pelo qual o YOLOv5 funciona tão bem com diferentes *datasets*.

3.2.2 *Alphapose*

Alphapose é um *pose estimator* estremamente preciso, capaz de detetar múltiplas pessoas em tempo real [8]. Este consegue prever e fazer rastreamento da localização de um objeto ou pessoa. Tudo isto é feito através da combinação da pose e da orientação.

O problema inerente em *pose estimation* é determinar a posição e orientação da câmara relativo ao objeto ou pessoa. Isto é tipicamente feito através da identificação, localização, e rastreamento de um número de pontos chave num objeto/pessoa. Para objetos, podem ser cantos ou outros traços importantes, para pessoas, estes pontos-chave representam as articulações, como por exemplo, um cotovelo ou joelho. O objetivo dos modelos de *machine learning* é conseguir rastrear estes pontos chaves em imagens ou vídeos.

3.2.3 *DeepSORT*

É um algoritmo de rastreamento por deteção que considera ambos os parâmetros da caixa delimitadora que advém da deteção e a informação sobre a aparência dos objetos rastreados, de forma a associar a deteção dos objetos numa nova *frame* com os previamente detetados [9] [10].

Simple Object Realtime Tracking (SORT) tem como principal objetivo fazer rastreamento de um objeto de uma única classe, em vez de múltiplos objetos. É também muitas das vezes referido como *Visual Object Tracking*. A caixa delimitadora, caixa que rodeia o objeto, é apenas definida na primeira *frame*, de forma manual, e o algoritmo irá ter que localizar o mesmo objeto no resto das *frames*, de forma automática. Como tal apenas considera a informação sobre as *frames* atuais e anteriores para conseguir realizar previsões sobre a atual *frame* sem ter que processar o vídeo todo.

3.3 Anotações

3.3.1 Fluxos de Anotações de *Frames*

O *script* carrega o vídeo e realiza a captura das *frames* através de dois métodos:

1. Captura de todas as *frames*
2. Captura de *frames* nas iterações definidas (ex: captura de uma frame a cada 8 frames)

Este último irá depender do valor SCALE_FACTOR. O vídeo será também passado pelos modelos de deteção e rastreamento de objetos *Yolov5* e *DeepSort*, obtendo um novo vídeo e um ficheiro .txt com todas as deteções e identificações de pessoas possíveis de se realizar. Neste primeiro passo obtemos apenas as *body boxes* de cada pessoa, obtendo as *head boxes* no processos seguintes. O ficheiro .txt criado, para cada registo, irá ter:

1. *ID* da *frame*
2. *ID* da pessoa
3. Coordenadas de rastreamento

Por fim, o novo vídeo é divido em *frames*. A figura 3.1 exibe o processo explicado.

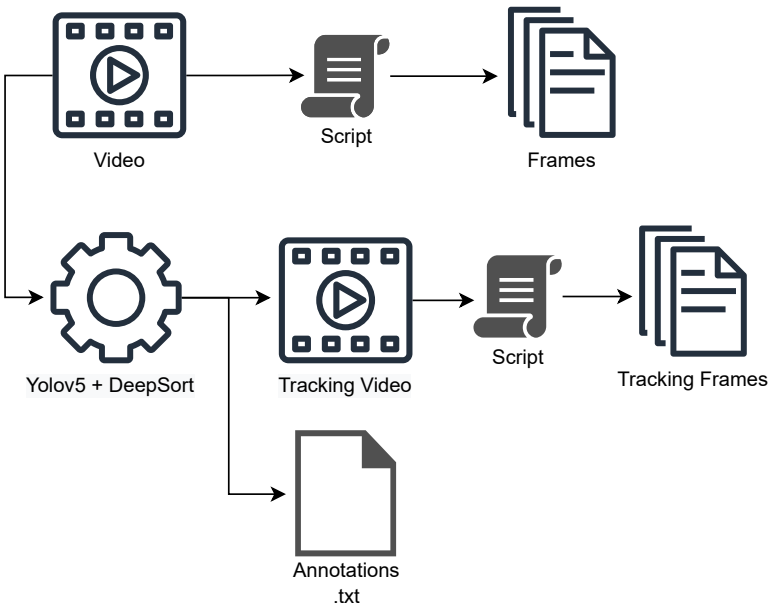

Figura 3.1: Divisão do vídeo em frames.

O *script* permite-nos corrigir os *IDs* de todas as pessoas anotadas no processo anterior, através do preenchimento da lista VALID_ID_LIST preemptivamente com os novos *IDs* e através da alteração da lista FIX_LIST com o intervalo de *frames*, o *ID* a atualizar e o novo. Por fim, o ficheiro .txt de anotações irá ser atualizado com as alterações realizadas.

De seguida, utilizando as *frames* obtidas no método 1 e o ficheiro .txt, é criado um novo ficheiro com a conversão das anotações do ficheiro .txt, no formato .xml. Iremos utilizar o nome de cada *frame* de modo a identificar na qual cada anotação pertence.

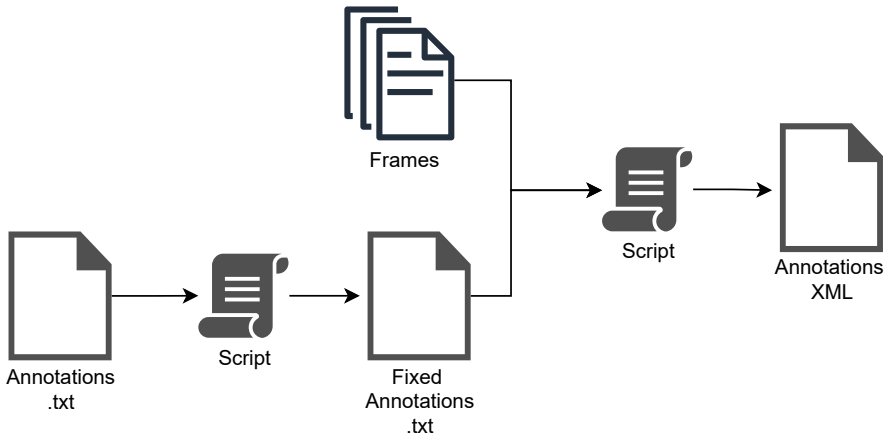

Figura 3.2: Criação do ficheiro de anotações no formato .xml.

Preparar a ferramenta *Computer Vision Annotation Tool* (CVAT), que nos permite fazer *upload* de *frames* e realizar manualmente todas as possíveis identificações e anotações de pessoas. Neste caso, iremos fazer *upload* das *frames* do método 2 e do ficheiro com anotações no formato .xml originado no *script* anterior e apenas corrigir as anotações necessárias e extrair o ficheiro .xml resultante.

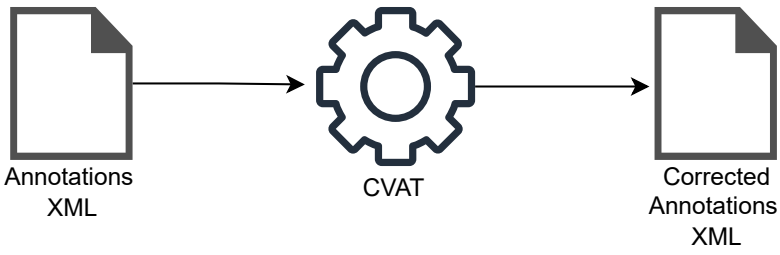

Figura 3.3: Procedimento no CVAT.

O *script* através das *frames* obtidas no método 2 e do ficheiro com anotações no formato .xml obtido no *script* anterior, cria um ficheiro com a conversão das anotações no ficheiro .xml em anotações no formato .json e gera novas com as respetivas *body boxes*. De seguida este ficheiro .json e as *frames* geradas anteriormente são passadas pelo *pose estimator Alphapose*, obtendo um novo ficheiro .json com o *output* resultante.

Por fim no ficheiro com as anotações no formato .json são introduzidas as anotações das *head boxes*, utilizando as frames com as respetivas *body boxes*, as anotações no formato .json e o ficheiro de output resultante do *Alphapose*.

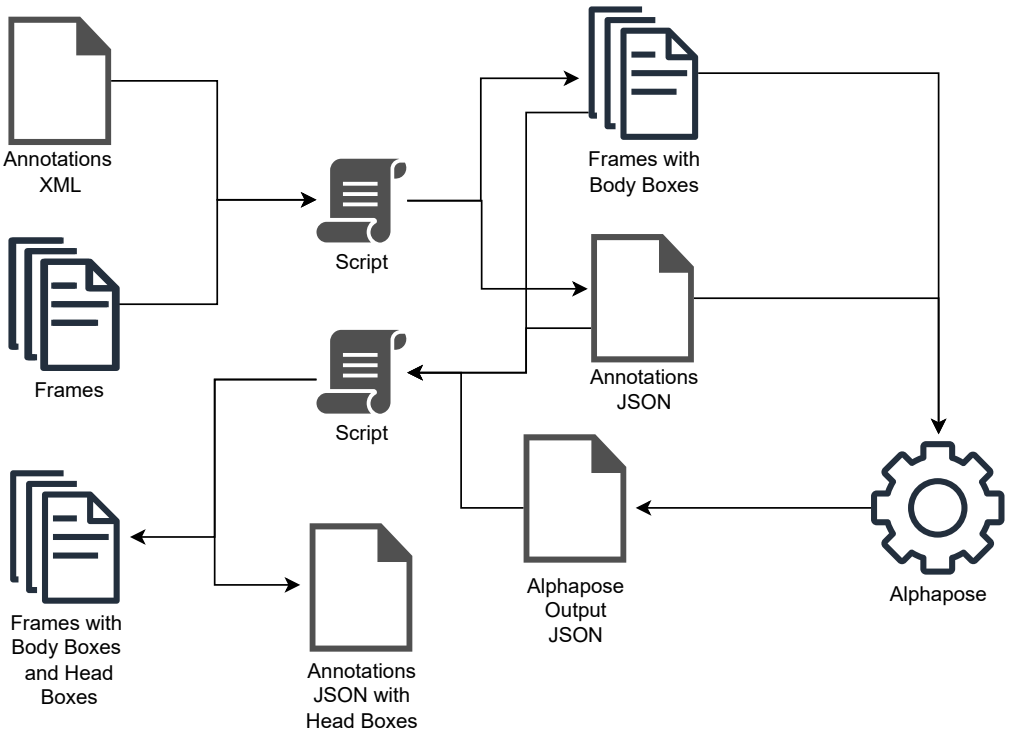

Figura 3.4: Utilização do *pose estimator Alphapose*.

O *script* introduz as anotações das *head boxes* no ficheiro com anotações no formato .xml tendo como *input* as *frames* obtidas no método 1, o ficheiro com anotações no formato .xml e o ficheiro com anotações no formato .json. Iremos utilizar o nome de cada *frame* de modo a identificar na qual cada anotação pertence.

Desta forma, através da atualização do ficheiro .xml com as anotações das *head boxes*, vamos utilizar de novo a ferramenta CVAT com as *frames* do método 2 para corrigir quaisquer problemas nas *head boxes*.

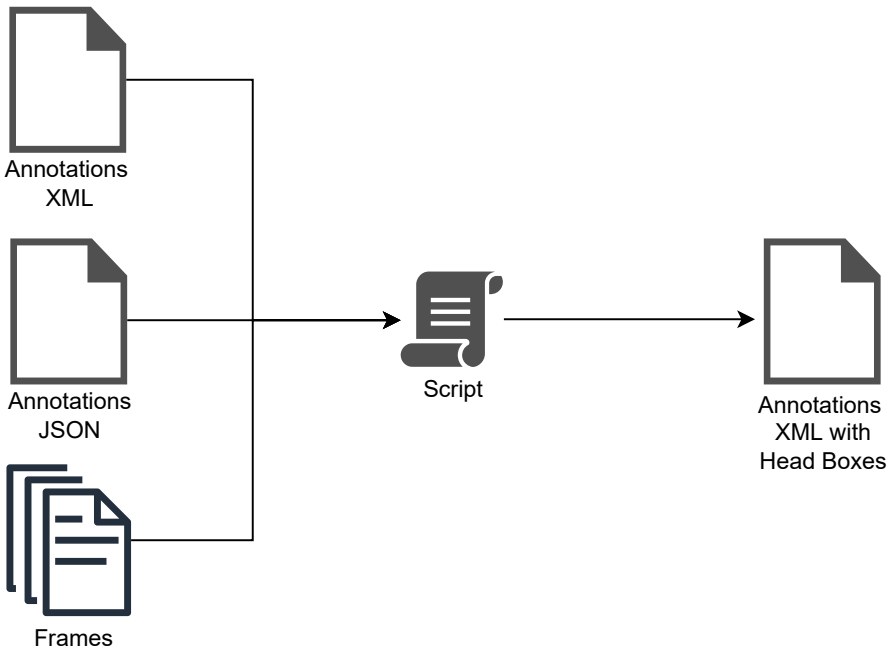

Figura 3.5: Obtenção das anotações no formato .xml com as respetivas *head boxes*.

O *script* através dos ficheiros com anotações no formato .json e .xml, atualiza o ficheiro .json com as respetivas *head boxes*.

De seguida, usando o ficheiro .json atualizado e as *frames* do método 1, vai ser originado o ficheiro .json atualizado com a inferência de anotações nas *frames* não anotadas, visto que, as anotações foram realizadas inicialmente sobre as *frames* do método 2.

Por fim, atualizamos o ficheiro com anotações no formato .xml com as anotações inferidas, fornecendo como input o ficheiro .json atualizado anteriormente, as *frames* do método 1 e o ficheiro .xml que iremos atualizar.

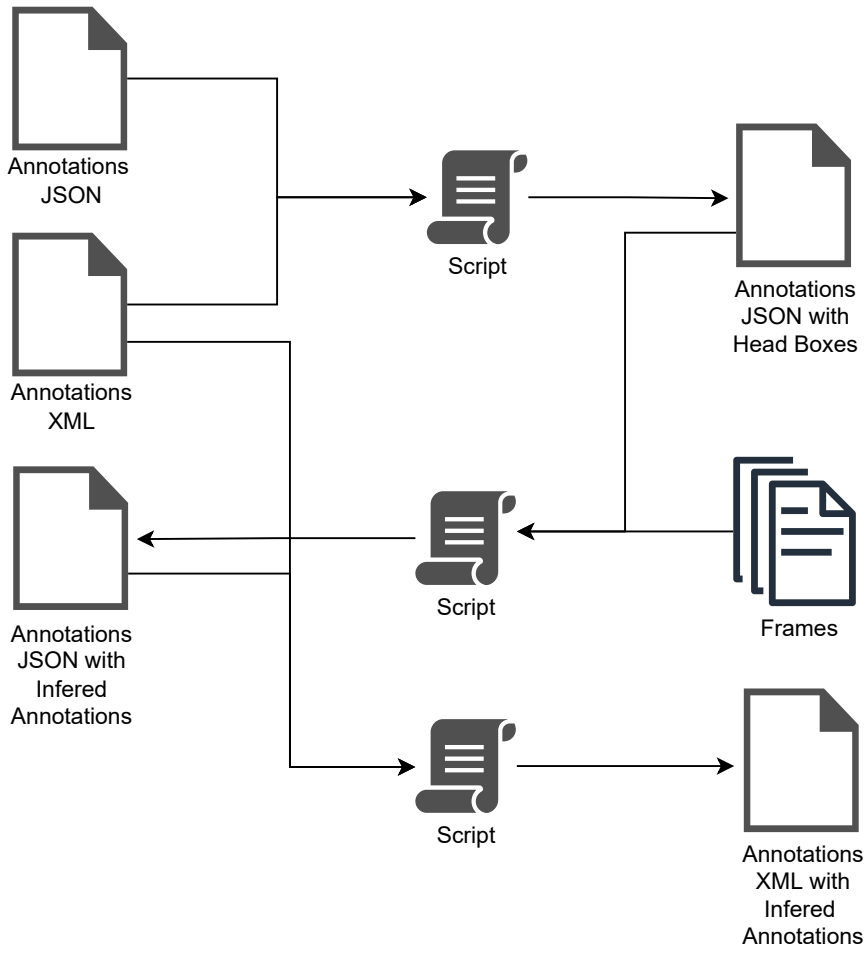

Figura 3.6: Obtenção das anotações inferidas no formato .xml.

O *script*, usando o ficheiro com anotações no formato .json e as *frames* do método 1, irá criar novas *frames* com as respetivas *head* e *body boxes* e criar um vídeo final com as todas as anotações. Para criarmos este novo vídeo, iremos preencher as *frames* do método 1 com as respetivas anotações e de seguida vamos "construir"um novo vídeo usando estas *frames*, dando uso á função *vídeoWriter* da biblioteca OpenCV.

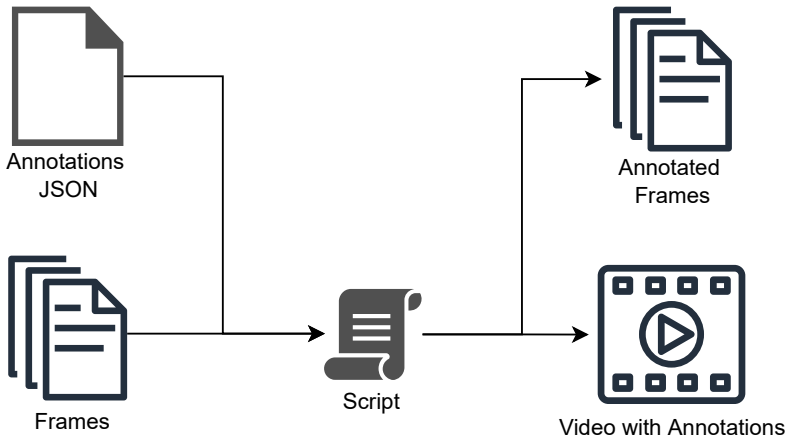

Figura 3.7: Criação do vídeo final com as anotações.

3.3.2 Fluxo de Anotações de Fala

O *script* recebe como *input* o vídeo original e irá reproduzir este, mas antes de o executar é necessário identificar o número de pessoas que irão falar no vídeo e mapear a identificação de cada uma destas a uma determinada tecla. A velocidade do vídeo, ou seja, o compasso de espera entre cada *frame* vai ser calculada dividindo variável *"multiplier"* pelo número de *Frames per Second* (FPS) e a velocidade do áudio será calculada pelo inverso do *"multiplier"*.

O fluxo de ações necessário para capturar as anotações de fala num vídeo em análise passa por carregar na tecla mapeada a uma determinada pessoa quando esta inicia a fala, assim que esta terminar de falar clica-se nessa mesma tecla indicando assim um período de fala num intervalo de frames. De notar que esta indicação apenas deve contemplar fala e não todos os sons emitidos pelas pessoas.

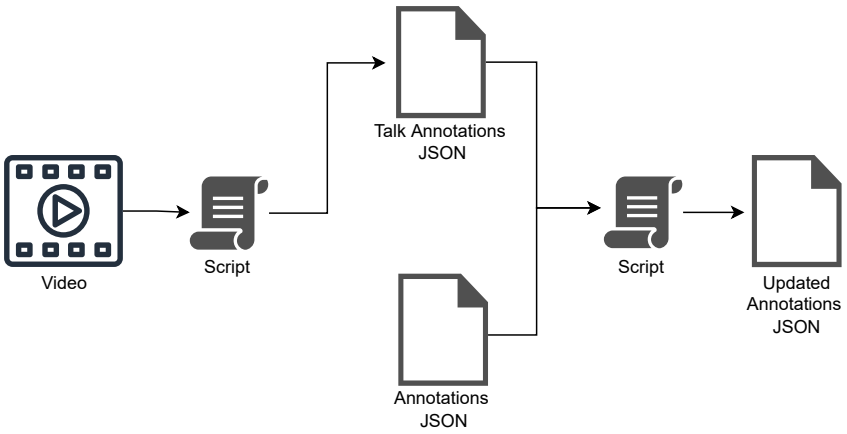

Figura 3.8: Atualização do vídeo final com as anotações de fala.

3.3.3 Fluxo de Conversão de Anotações de *Frames* para o formato AVA

De forma a conseguirmos utilizar as nossas anotações no modelo de ASD teremos que as converter para o formato necessário. O *script*, usando o ficheiro com anotações manuais no formato .json e o vídeo correspondente, irá criar dois novos ficheiros .csv. Estes são a conversão das anotações para o formato usado no *dataset Atomic Visual Action* (AVA) (val_orig.sh), e para o formato usado no *dataset LJSpeech* (val_loader.sh). No nosso caso precisamos apenas do ficheiro val_orig.sh, que irá será depois utilizado no modelo estado de arte escolhido.

Explicando de forma mais elaborada, nos ficheiros .json o formato das anotações guardadas pode ser representado da seguinte forma: a cada *frame* é lhe associado um *id* e um nome, as anotações têm também um *id* associado, contém o *id* da *frame* e da pessoa correspondente, contém o dados da *bounding box* da pessoa, onde são guardados a localização e limites, e tem o registo se esta está ou não a falar. O armazenamento das anotações neste formato facilita o trabalho de manipulação dos ficheiros, tendo fácil acesso a qualquer informação específica. No entanto, o grande problema de termos as anotações neste formato é que vão originar ficheiros de grandes dimensões, como tal alguns dos modelos optam por utilizar ficheiros *Comma-separated values* (CSV) que compactam a informação. Na conversão dos .json para .csv, o novo formato de anotações irá ter: o nome identificador do conjunto ao qual o vídeo em questão pertence, um valor indicador do momento temporal no vídeo em que a *frame* aparece, onde este começa a zero e vai sendo cal-

culado através da divisão do número de *frames* pelo número de FPS, quatro valores de coordenadas que indicam a posição e limites da *bounding box*, o registo se a pessoa esta ou não a falar, que irá ter o valor *SPEAKING* ou *NOT SPEAKING*, e o nome do vídeo juntamente com o número 1 ou 0, que represente se a pessoa está a falar ou não, respetivamente. O resultado final, como já foi mencionado acima, será um único ficheiro .csv (val_orig.sh) com todas as anotações de um conjunto de vídeos.

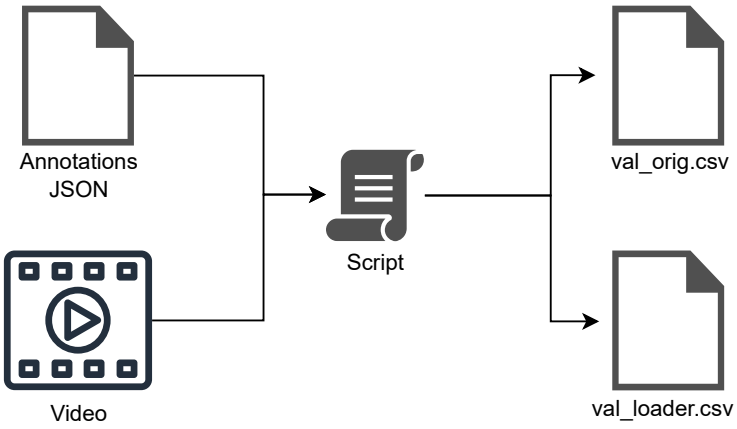

Figura 3.9: Conversão de anotações para formato AVA.

3.4 Conclusões

Neste capítulo foi apresentado todo o processo que envolveu a implementação deste projeto, desde as ferramentas, à implementação dos processos de anotação de *frames* e conversão destas para o formato AVA, e implementação do processo de anotação de fala. A anotação de *frames* foi, como já foi mencionado, realizada de forma automática dando uso ás ferramentas mencionadas e as anotações de fala foram feitas de forma manual. Alguns destes processos sofreram algumas alterações ao longo do projeto, de forma a facilitar estes e acelerar a obtenção dos resultados.

Capítulo

4

Avaliação do modelo Active Speakers in Context

4.1 Introdução

Neste capítulo abordamos a testagem e análise do modelo estado de arte escolhido para ASD. Este capítulo foi dividido em vários pontos de interesse, sendo eles: o modelo escolhido, razões pelo qual este foi escolhido, descrição dos conjuntos de dados utilizados e discussão sobre os resultados obtidos.

4.2 *Active Speakers in Context*

De entre os modelos estado de arte considerados para avaliação (ASC[11], ASDNet e *Multi-Modal Assignation for Active Speaker Detection* (MAAS)), o ASC foi o escolhido pois foi o único que conseguimos correr sem problemas significativos, ao contrário dos outros, que apresentavam falhas por parte dos autores, como o não fornecimento de algumas das ferramentas necessárias para o conseguirmos correr, o que nos inibiu de realizar a sua implementação. Será também um bom modelo de referência ao teste dos atuais modelos estado de arte para ASD, visto que, utiliza processos muito semelhantes aos que estes implementam.

O ASC divide-se em dois processos principais, o *Short-Term Encoder* (STE) e o ASC. O primeiro é composto por duas redes: uma que extraí uma representação dos recortes das caras das pessoas num determinado intervalo de tempo e converte o áudio num *Mel-Spectogram* (normalmente utilizado em aplicações de classificação de áudio), e a outra rede que obtém uma representação do áudio no mesmo intervalo de tempo mencionado anteriormente.

Este primeiro processo serve para extrair as características das imagens das caras e do áudio fornecidos ao modelo. Através deste primeiro processo, o modelo consegue prever quem está a falar num conjunto de vídeos fornecidos.

A segunda parte do modelo consiste em utilizar as relações entre as várias pessoas num intervalo de tempo de forma a melhorar a inferência de quem está a falar, através da construção de conjuntos de contexto, onde temos a pessoa para qual queremos analisar a sua atividade de fala (pessoa de referência) e as restantes pessoas que também falam nesse intervalo de tempo (falantes de contexto). Este processo é feito tendo em conta um intervalo de tempo superior ao que o modelo previamente analisou. O modelo associa ao intervalo de tempo as representações obtidas no processo anterior, e tendo a pessoa escolhida como referência, se existirem outras que estejam a falar nesse mesmo instante são consideradas como contexto e guardadas em conjunto com a representação da pessoa de referência. Este processo é repetido para todas as pessoas em diferentes intervalos de tempo ao longo do vídeo.

Quando todo o conjunto de contexto estiver conntruído este será melhorado através de dois processos, *Pairwise Refinement* e *Temporal Refinement*. O primeiro, *Pairwise Refinement*, é realizado de forma a extrair as relações entre as representações das várias pessoas independentemente da ordem temporal em cada conjunto de contexto. De seguida o *Temporal Refinement*, tem como função determinar a importância das relações extraídas tendo em conta a posição das características de cada pessoa no intervalo de tempo. Realiza também o tamanho das representações para a camada de previsão. Neste dois passos é utilizado o modelo *Long short-term memory* (LSTM), em que o *output* irá ter dimensão necessário para ser passado pela camada de previsão. Por fim, em cada intervalo de tempo dos vários conjuntos de contexto o modelo prevê se a pessoa de referência está a falar ou não.

4.3 Conjuntos de Dados

Temos dois conjuntos de dados, o AVA-*ActiveSpeaker* e o Wild Active Speaker Detection (WASD). O AVA-*ActiveSpeaker* é constituído por 38 horas já o *WASD* conjunto de dados anotado, *WASD*, tem 30 horas distribuidas por 5 categorias. Estas categorias são: Debate, *Interview*, *React*, *Podcast* e *Police*. Os vídeos escolhidos que compõem o WASD, permite-nos obter proporções equilibrada para as características, Linguagem, Raça e Género, O AVA-*ActiveSpeaker* foi utilizado para treinar a avaliar o *WASD* modelo estado de arte, enquanto que o WASD serve apenas para avaliação, visto que, não é grande o suficiente para treinar o modelo.

De forma a melhorar a variedade do *WASD* e testar a eficácia do modelo em questão, aplicamos as características Linguagem, Raça e Género, na escolha dos nossos vídeos. Esta distribuição é feita pois, por exemplo, na cultura asiática quando uma pessoa está a falar é comum haver pequenas interjeições por parte de outras, especialmente no diálogo entre mulheres. Este fenómeno também se demonstra em cenários com afro-americanos, também predominantemente entre mulheres, já em vídeos com pessoas caucasianas não ocorre com tanta frequência. Toda esta divisão ajuda, visto que, o AVA-*ActiveSpeaker* é constituído por filmes de *Hollywood* onde são a pessoas são maioritariamente caucasianas.

O modelo irá ter mais dificuldade em inferir de forma correta com o nosso conjunto de dados devido a uma maior variedade de situações como o *overlapping* de comunicação. Outro fator a ter em conta é que no AVA-*ActiveSpeaker*, sendo composto por filmes, o diálogo destes é premeditado, logo tem maior coerência do que em cenários do dia-a-dia. A distribuição demonstrada nos gráficos seguintes demonstra o equílibro atingido quanto ás proporções das características do dados que compõem o *WASD*.

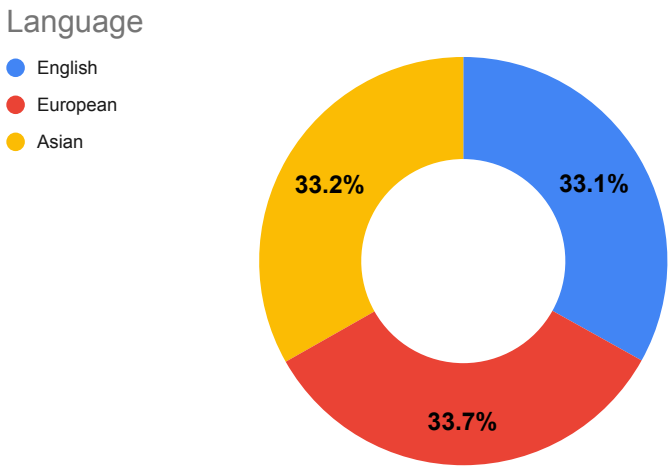

Figura 4.1: Proporções da característica Linguagem.

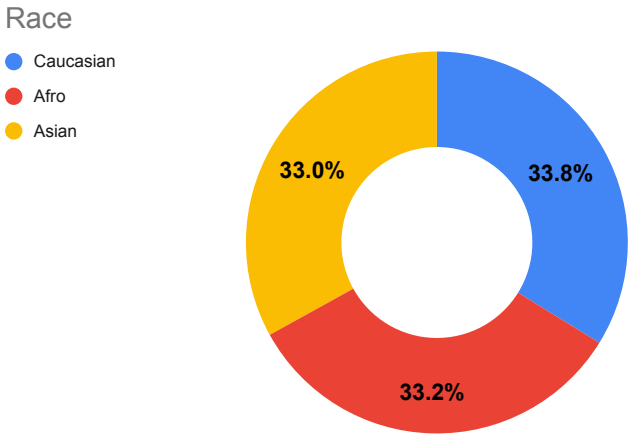

Figura 4.2: Proporções da característica Raça.

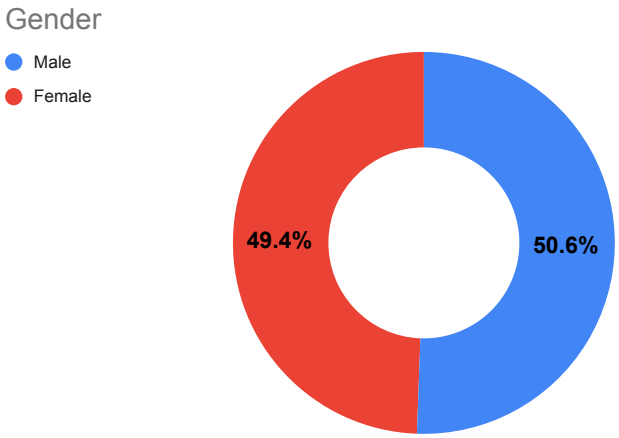

Figura 4.3: Proporções da característica Género.

As categorias apresentam diferentes níveis de acesso às bases necessárias para inferência, sendo estas o acesso às caras, e a qualidade de imagem e som. Através disto é possivel obter um equílibrio de qualidade no *WASD*, tendo vídeos com diferentes graus de dificuldade para o modelo. Isto permite-nos realizar uma testagem mais realista do modelo ao contrário do AVA-*ActiveSpeaker* que, como já foi mencionado, consiste em diversos filmes de *Hollywood* anotados, que como tal trata-se de cenários caracterizados por boa iluminação, bom acesso á(s) cara(s), e boa qualidade de som. A figura seguinte representa as categorias e os seus níveis de acesso às caras e áudio.

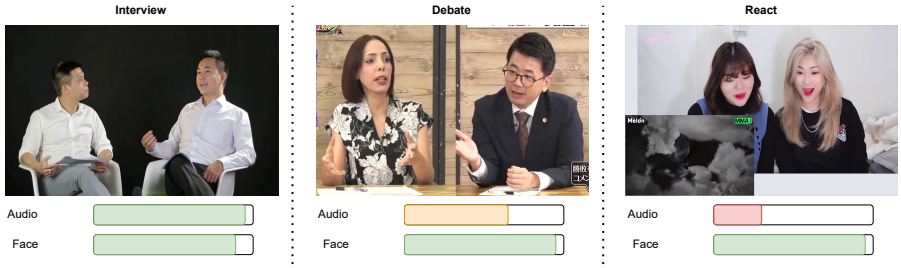

Figura 4.4: Níveis de dificuldade de acesso ao áudio e cara para as categorias *Interview*, *Debate* e *React*

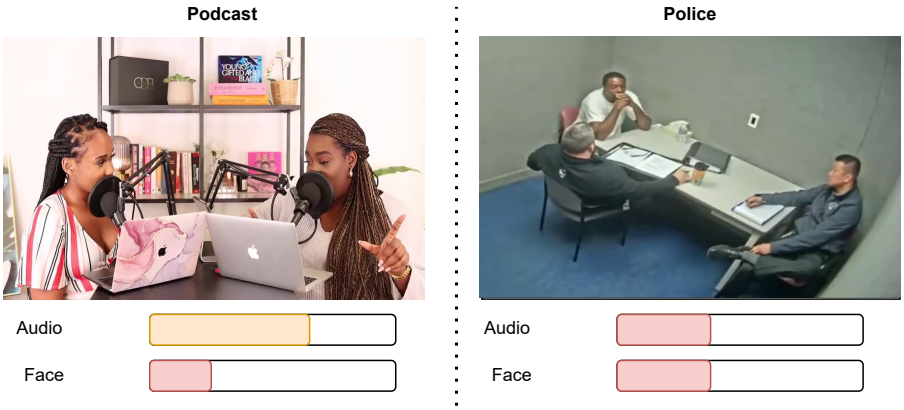

Figura 4.5: Níveis de dificuldade de acesso ao áudio e cara para as categorias *Podcast* e *Police*

Devido às diferentes categorias escolhidas para representar o WASD, calculamos as proporções cara/corpo no geral em cada uma. Através destes valores iremos conseguir verificar as diferentes dificuldades que modelo irá ter no acesso às caras. Os valores obtidos encontram-se na seguinte tabela.

Categoria	**Proporção cara/corpo**
Debate	0.7
Interview	0.4
Podcast	0.2
React	1.5
Police	0.9

Tabela 4.1: Proporções cara/corpo para cada categoria do WASD, resultando da análise dos ficheiros .json de um conjunto de vídeos

O modelo não tem o mesmo acesso às caras de categoria em categoria, não conseguindo encontrar padrões o que irá não só dificultar o seu trabalho, mas também irá influenciar os resultados obtidos em comparação ao AVA-*ActiveSpeaker*, onde estas proporções mantêm-se constantes ao longo de todo o conjunto de dados devido aos vídeos serem todos extraídos da mesma fonte, e como tal têm as mesmas características.

4.4 Resultados Obtidos e Discussão

Baseado no gráfico apresentado, é possível realizar uma estimativa dos valores para cada categoria, sendo que a categoria *Interview* será a que possui o valor *Mean Average Precision* (mAP) mais elevado e com maior facilidade de inferir, visto que, é a categoria que melhor ataca as características, som e imagem, e com melhor acesso à cara, pois esta irá estar sempre visível e com boa qualidade.

A pior categoria seria a *Police*, sendo o cenário mais *wild*, sem garantia de acesso a cara nem som com qualidade. Este cenário é o que diverge mais do AVA-*ActiveSpeaker* pelo que o modelo terá muita dificuldade em inferir, pois não têm acesso garantido a nenhuma das suas bases (cara e som). Isto deve-se pois a cara nem sempre é visível ou os vídeos não têm qualidade de imagem suficiente para perceber quem está a fala (cara distante e imagem com pouca qualidade) e o som nem sempre é o melhor (som distante e com eco).

Para efeitos de teste e comparação futura, iremos ter o modelo de duas maneiras, uma versão onde o modelo é treinado por nós mesmos, e a versão pré-treinada. Na primeira versão foi efetuada em primeira instância o treino dos processos *STE* e *ASC* usando o AVA-*ActiveSpeaker*, e por fim realizamos o *Forward* e *Post-Processing* de cada processo. Na segundo versão apenas precisamos de realizar o *Forward* e *Post-Processing* dos processos. De seguida, usando o *WASD* como conjunto de validação e o AVA-*ActiveSpeaker* como conjunto de treino, realizamos o *Forwarding* e *Post-Processing* de cada processo já treinado anteriormente, ou seja, utilizamos a versão pré-treinada do modelo. Os valores obtidos em ambos os *datasets* estão na seguinte tabela.

Dataset	**ASC**
AVA-ActiveSpeaker	80.2 (87.1)
Wild Active Speaker Detection	35.3

Tabela 4.2: Resultados da avaliação do AVA-*ActiveSpeaker* e do WASD. O valor mAP entre parênteses é o valor reportado pelos autores.

O nosso valor obtido no AVA-*ActiveSpeaker* aproxima-se do valor reportado pelos autores, demonstrando que todo o processo foi bem sucedido. Face à toda a advertência imposta sobre os vídeos que compõem o *WASD*, o valor *mAP* obtido está de acordo com o expectado. Este quando comparado com o valor obtido através do AVA-*ActiveSpeaker* onde possui vídeos com melhor qualidade de imagem, áudio e melhor acesso às caras que facilita o trabalho do modelo, é possível concluir que o resultado obtido para o *WASD* demonstra que o modelo apresenta dificuldade em realizar ASD com um conjunto de dados composto por vídeos num ambiente *wild*, ou seja, num ambiente não controlado, como previsto.

4.5 Conclusões

Neste capítulo foi apresentada a testagem e análise do modelo estado de arte para ASD escolhido. Foi mencionada e discutida a razão para o qual escolhemos este modelo, mostramos e descrevemos as características de ambos os conjuntos de dados utilizados para avaliação do modelo e por fim, apresentamos e discutimos os resultados obtidos das testagens feitas ao modelo usando ambos os conjuntos de dados, onde apenas alternamos os conjuntos de validação entre o nosso e o do AVA-*ActiveSpeaker*, estando estes resultados de acordo o expectado.

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Este projeto teve como objetivo principal estudar, implementar, testar e avaliar o modelo ASC para ASD, utilizando o *dataset* em conjunto com o nosso próprio conjunto de dados, WASD. A implementação do modelo escolhido permitiu-nos verificar que existe problemas nos modelos existentes, nomeadamente a falta do conteúdo necessário para correr alguns destes o que não permite realizar qualquer tipo de testagem relevante.

Os resultados obtidos da avaliação do ASC, utilizando o AVA-*ActiveSpeaker* e o nosso próprio conjunto de dados, WASD, permitiu-nos confirmar a motivação por trás deste projeto, em que os modelos estado de arte atuais para ASD apresentam diversas limitações quando expostos a ambientes *wild*, sendo que estes são mais realistas e estão mais de acordo ao que um modelo poderá ser exposto quando utilizado num ambiente real, como por exemplo, num sistema de videovigilância.

5.2 Trabalho Futuro

Para comprovar as limitações do modelo, poderíamos correr este com cada categoria como sendo um *dataset* individual. Isto iria ajudar não só a verificação das bases mais importantes para o modelo, mas também o sedimentar da conclusão obtida, onde foi possível observar que o modelo não está preparado para um *dataset wild*, sendo possível obter melhores resultados com um *dataset* controlado, onde os vídeos apresentam um bom acesso às caras, bem como boa qualidade de imagem e som. No futuro é possível também testar combinações de categorias de forma a verificar quais as bases mais im-

portantes para o modelo, como já foi mencionado. Poderia ser realizada a combinação *Interview* + *Debate*, onde difere do *dataset* AVA contudo mantem as mesmas características que os modelos estado-da-arte necessitam para inferir quem está a falar (acesso á cara e som). Outra combinação possível seria *React* + *Podcast* que ataca os dois pontos críticos dos modelos de forma cirúrgica, som e cara, respetivamente, onde existe uma maior qualidade de imagem e som devido ao posicionamento relativo à câmara.

Bibliografia

[1] Chong Huang and Kazuhito Koishida. Improved active speaker detection based on optical flow. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[2] Ondrej Klejch Radhika Marvin Andrew Gallagher Liat Kaver Sharadh Ramaswamy Arkadiusz Stopczynski Cordelia Schmid Zhonghua Xi Caroline Pantofaru Joseph Roth, Sourish Chaudhuri. Ava-activespeaker: An audio-visual dataset for active speaker detection. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[3] You Jin Kim, Hee-Soo Heo, Soyeon Choe, Soo-Whan Chung, Yoohwan Kwon, Bong-Jin Lee, Youngki Kwon, and Joon Son Chung. Look who's talking: Active speaker detection in the wild. In *Interspeech*, 2021.

[4] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 3927–3935, 2021.

[5] Ruohan Gao and Kristen Grauman. Visualvoice: Audio-visual speech separation with cross-modal consistency. In *CVPR*, 2021.

[6] Ali K. Thabet Bernard Ghanem Juan Leon Alcázar, Fabian Caba Heilbron. Maas: Multi-modal assignation for active speaker detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[7] Glenn Jocher. YOLOv5, 2020. [Online] `https://github.com/ultralytics/yolov5`. Último acesso a 7 de julho de 2022.

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, 2017.

[9] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[10] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018.

[11] Long Mai Federico Perazzi Joon-Young Lee Pablo Arbelaez Juan Leon Alcázar, Fabian Caba Heilbron and Bernard Ghanem. Active Speakers in Context, 2020. [Online] `https://github.com/fuankarion/active-speakers-context`. Último acesso a 7 de julho de 2022.