Universidade da Beira Interior
Departamento de Informática

Nº 129 - 2022: *Avaliação de Active Speaker Detection in Wild Conditions I*

Elaborado por:

João Bernardo Moroso Benquerença

Orientador:

Professor Doutor Hugo Proença

8 de julho de 2022

Agradecimentos

A conclusão deste trabalho, bem como a realização do mesmo não seria possível sem o acompanhamento feito pelo Professor Hugo Proença e pelo Professor Tiago Roxo ao longo do semestre.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ASC *Active Speaker Context*
AVA *Atomic Visual Actions*
CVAT *Computer Vision Annotation Tool*
FPS *Frames per Second*
JSON *JavaScript Object Notation*
IOU *Intersection over Union*
mAP *mean Average Precision*
NMS *NonMaximum-Suppression*
RMPE *Regional Multi-person Pose Estimation*
SDTN *Spatial Detransformer Network*
SORT *Simple Online and Realtime Tracking*
SPPE *Single-Person Pose Estimator*
STE *Short-Term Encoder*
STN *Spatial Transformer Network*
UBI Universidade da Beira Interior
XML *Extensible Markup Language*
YOLO *You Only Look Once*

Capítulo

1

Introdução

1.1 Enquadramento

Este projeto foi desenvolvido no âmbito da unidade curricular de projeto do curso de Engenharia Informática da Universidade da Beira Interior (UBI). Consiste na anotação de um conjunto de vídeos, avaliação de um modelo de estado de arte para *active speaker detection* utilizando o conjunto *Atomic Visual Actions* (AVA) *Active Speaker Dataset* e um conjunto anotado ao longo do semestre, e por fim uma discussão dos resultados obtidos. Na tarefa de *active speaker detection*, um modelo deteta ao longo de um vídeo quem e quando é que alguém está a falar.

1.2 Motivação

O projeto surge com o propósito de testar um modelo de estado de arte para *active speaker detection* num conjunto de vídeos *wild*, que se passam em situações de vida real. Ao contrário do AVA *Active Speaker Dataset*, conjunto de dados de referência para esta tarefa, e onde os vídeos são tirados de filmes.

1.3 Objetivos

Os objetivos do projeto são, começar por analisar um dos modelos de estado de arte para *active speaker detection*, e replicar os resultados reportados pelos autores do modelo no AVA *Active Speaker Dataset*. O segundo objetivo é anotar um conjunto de vídeos e de seguida adaptar este para o formato do conjunto AVA. Depois, avaliar o modelo neste conjunto anotado e perceber

quais os elementos que fazem o desempenho do modelo descer. Por último, a escrita do relatório.

1.4 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresentação do projeto, motivação para a sua escolha, enquadramento para o mesmo e organização do relatório;

2. O segundo capítulo – **Estado de Arte** – descrição de sete modelos de estado de arte para *active speaker detection*;

3. O terceiro capítulo – **Tecnologias e Ferramentas Utilizadas** – aborda as tecnologias e ferramentas utilizadas no desenvolvimento dos programas para fazer as anotações;

4. O quarto capítulo – **Anotações** – explicação do processo para fazer e converter para o formato AVA as anotações;

5. O quinto capítulo – **Avaliação do modelo *Active Speakers in Context*** – avaliação do modelo de estado de arte escolhido no conjunto de dados anotado;

6. O sexto capítulo – **Conclusões e Trabalho Futuro** – conclusões principais tiradas da realização do projeto e trabalho futuro.

Capítulo

2

Estado da Arte

2.1 Introdução

Este capítulo serve para apresentar vários modelos de estado de arte utilizados para a tarefa de *active speaker detection*.

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

Este artigo introduz, para além do conjunto de dados AVA-*Active Speaker Dataset*, um modelo para a tarefa de *active speaker detection* [1]. Para esta tarefa é preciso fazer uma análise conjunta de características visuais e de áudio, em que o objetivo do modelo é encontrar uma função em que dados um conjunto de imagens com recortes de caras e um segmento de áudio é devolvida uma probabilidade da pessoa nessas imagens estar a falar. A estrutura do modelo consiste de duas redes que obtém uma a representação do áudio e a outra das imagens, que são depois dadas a uma última parte que as vai juntar e obter as previsões. Para as primeiras duas redes foram utilizadas redes neuronais convolucionais. Na última parte de modelo, os autores do artigo tentaram dois métodos diferentes, sendo que aquele com que obtiveram melhores resultados foi uma rede neuronal recorrente, utilizada para fazer a fusão das representações anteriores. Destaca-se por conseguir manter informação à medida que se processa a entrada da rede, conseguindo relacionar saídas da rede em instantes anteriores com os dados a serem processados no momento atual. Depois de se obter a fusão, esta é passada a duas camadas *softmax* que vão devolver, para cada pessoa, a probabilidade de num determinado instante do

vídeo estar a falar.

2.3 Active Speakers in Context

Este modelo divide-se em duas partes principais, o *Short-Term Encoder* (STE) e o *Active Speaker Context* (ASC) [2], o primeiro serve para extrair as características das imagens das caras e do áudio, e é composto por duas redes. Os recortes da cara são dados a uma rede que devolve uma representação destas e o áudio é convertido para um *Mel-spectogram* e é dado a outra rede que obtém uma representação do áudio referente a um certo intervalo de tempo. O resultado das duas é depois concatenado e com isto, o modelo já consegue fazer uma previsão de quem está a falar.

A segunda parte do modelo foca-se em construir, e melhorar, um contexto para a pessoa para quem está a fazer a previsão, de forma a identificar relações com as outras pessoas que permitam indicar com mais certeza que a pessoa está a falar. Isto é feito analisando um intervalo de tempo superior ao que o modelo está a processar para ser possível olhar para o que estava a acontecer antes e depois daquele momento. O modelo coloca por cima do intervalo de tempo as representações obtidas anteriormente, e seleciona a pessoa para quem está fazer a previsão como referência. Cada representação é obtida para um determinado instante para uma determinada pessoa, se no instante em que o modelo está a analisar houverem outras que se sobreponham à pessoa de referência essas são consideradas como contexto e guardadas em conjunto com a representação da pessoa de referência. Este processo é depois repetido para todas as pessoas no vídeo em vários intervalos de tempo ao longo do vídeo. Depois de construir o contexto são feitos dois processos para o melhorar. O primeiro, *Pairwise Refinement*, é feito para extrair as relações entre os vetores de contexto de duas de pessoas de cada vez, independentemente da ordem temporal. Depois, é feito o *Temporal Refinement*, que tem a função de determinar a importância das relações extraídas tendo em conta a sua posição no intervalo de tempo. Tem também a função de diminuir o tamanho dos vetores de contexto para a camada de previsão. Nesta, em cada instante a partir dos vetores de contexto o modelo prevê se a pessoa de referência está ou não a falar.

2.4 Improved Active Speaker Detection based on Optical Flow

Os modelos utilizados para *active speaker detection* são muitas vezes afetados com vários problemas que dificultam a previsão como movimentos de uma pessoa que não está a falar, variações na iluminação e pouca definição no vídeo. Este modelo utiliza uma técnica chamada *optical flow* para tentar que estes problemas não o afetem tanto [3]. O *optical flow* é uma representação do movimento e pode ser medido utilizando as diferenças entre duas imagens seguidas. Traz vantagens como conseguir ilustrar o padrão que os movimentos da boca da pessoa que está a falar faz, conseguir detetar movimentos subtis na cara das pessoas e mesmo com variações na iluminação o *optical flow* gerado é consistente e consegue evitar que a rede aprenda correspondências incorretas entre a pessoa a falar e a anotação de quem está a falar.

O modelo processa separadamente as entradas visuais e de áudio, para as entradas visuais são propostos dois métodos. O primeiro é o *Visual-Coupled Embedding* e o segundo é o *Independent Embedding*. No primeiro as imagens das caras e o *optical flow* são dados juntos, isto permite à rede aprender relações entre as imagens e os *optical flows*, conseguindo extrair informação importante entre a aparência da cara e o movimento. No segundo método as imagens das caras e os *optical flows* são dados a duas redes diferentes e o resultado destas é concatenado no final. A saída das duas estratégias consiste num vetor que representa características das imagens e *optical flows*. Para obter a representação do áudio é dada a uma rede uma série de espectrogramas de Mel, que são obtidos a partir de um intervalo de tempo de áudio. A última camada do modelo é uma rede responsável por fazer as previsões, esta recebe os vetores com as representações do áudio e do vídeo concatenados. A rede é baseada num modelo *sequence-to-sequence*, em que a entrada passa por um *encoder* que a transforma num vetor, este é a representação final antes de passar à previsão e captura as relações entre as características do áudio e do vídeo. O vetor é depois passado ao *decoder* que produz uma sequência de previsões com determinada probabilidade.

2.5 How to Design a Three-Stage Architecture for Audio-Visual Active Speaker Detection in the Wild

Este modelo é constituído por três componentes [4], este servem para:

1. integração de informação audio-visual para cada pessoa;
2. informação contextual que captura a relação entre as pessoas no vídeo;
3. modelo de tempo para explorar relações de longo termo na conversa.

Ao primeiro componente são dados os recortes das caras das pessoas e o áudio de um vídeo. Cada um é processado por uma rede e este modelo distingue-se da maioria do outros modelos de estado de arte por utilizar o sinal original de áudio, sem o passar por nenhum filtro. Os vetores que saem das duas redes são concatenados e servem de entrada na próxima componente que vai extrair relações entre as pessoas no vídeo. Para fazer isto são concatenadas as características extraídas dos recortes da cara de cada pessoa e dadas a uma rede, exceto da pessoa para a qual o modelo está a fazer a previsão. O resultado desta rede é concatenado às características da pessoa em questão e o modelo passa então para o último passo. Onde vai ser analisado se a pessoa estava a falar no instante anterior ou se está a falar no instante seguinte ao que está a avaliado pelo modelo. A última camada devolve um vetor para cada instante, e em que cada posição corresponde a uma pessoa no vídeo, nessa posição o valor é igual a 1 se esta estiver a falar ou 0, caso contrário.

2.6 MAAS: Multi-modal Assignation for Active Speaker Detection

Para fazer a tarefa de *active speaker detection* este modelo utiliza redes *multi-modal graph neural networks* [5]. A primeira parte do modelo recebe vários recortes de caras e um *Mel-Spectogram* obtido através do sinal de áudio original, referentes a um determinado intervalo de tempo. Os dois são dados a duas redes diferentes e são devolvidos vetores com as características de cada pessoa no vídeo e um vetor para o áudio. Estes são depois passados ao próximo passo que se chama *Local Assignation Network*, onde o modelo constrói um grafo em que os nodos são estes vetores e estam todos ligados entre si. O objetivo desta rede é detetar se alguma pessoa está a falar, e caso esteja, avaliar qual tem mais probabilidade de o estar a fazer. De seguida, o *Temporal Assignation* é utilizado para melhorar a precisão da previsão, faz isto construindo um grafo centrado no instante a ser avaliado e contém nodos dos instantes anteriores e seguintes. O grafo faz a ligação entre os vetores com características do áudio e entre os vetores com características visuais, caso estes correspondam às mesmas pessoas. No último passo, é utilizada uma rede que vai construir um grafo em que as ligações entre os nodos é feita tendo em conta a

proximidade nas características guardadas nos vetores. As relações encontradas por esta rede são concatenadas com as relações do *Temporal Assignation* e são dadas à última camada responsável então por fazer a previsão de *active speaker detection*.

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

O modelo tem o nome de *TalkNet* e consiste de apenas duas componentes, uma para extrair as características dos recortes das caras e do áudio e a segunda para fazer as previsões de *active speaker detection* [6]. A primeira consiste de duas redes para obter as representação visuais e de áudio. A rede para processar os recortes contém um *Visual Frontend* que é utilizado para extrair as características e uma *Visual Temporal Network* para melhorar a representação no vetor e reduzir as dimensões. Para obter a representação do áudio um segmento de áudio é convertido num vetor de *Mel-frequency cepstral coefficients*, que dado ao *audio temporal encoder* é utilizado para extrair a representação do áudio por uma rede neuronal. Depois de obter ambas as representações o próximo passo é juntar as duas. Para isto, é utilizada uma camada chamada *attention layer* esta vai aprender como as características do áudio influenciam as características do vídeo, e vice-versa. Á medida que estes resultados vão sendo calculados, são concatenados e dados a outra camada, chamada de *self-attention layer*, que vai extrair relações dentro de cada vetor devolvido pela camada anterior, sendo que depois desta é possível indicar quais são as *frames* em que alguém está a falar.

2.8 UniCon: Unified Context Network for Robust Active Speaker Detection

Os modelos apresentados anteriormente utilizam apenas os recortes das caras das pessoas nos vídeos e o áudio para extrair as relações entre estas. Este modelo propõem introduzir o uso da posição e escala das caras das pessoas como contexto para perceber o foco de atenção de cada pessoa na cena [7]. O nome do modelo é *Unicon* e começa por processar os recortes de cara de várias *frames* durante um determinado intervalo de tempo para obter vários vetores com características dos recortes. É também obtido um vetor com a representação do áudio correspondente ao mesmo intervalo de tempo dos

recortes. O próximo passo é construir o *spatial context* para representar a posição e tamanho da cara de cada pessoa. De seguida, para cada pessoa na *frame* é construído um mapa diferente em que esta é identificada pela cor amarelo e o resto por azul. Depois, para identificar relações entre duas pessoas de cada vez, uma delas fica com a cor vermelha e a outra a cor verde. Por último estes mapas são dados a uma rede neuronal convolucional para obter uma representação final destes, que são então o *spatial context*. O *relational context* serve para perceber qual a relação de cada pessoa no vídeo com a pessoa para a qual está a ser feita a previsão, para isto o modelo vai obter um vetor para contexto apenas visual e outro para contexto audio-visual. O primeiro serve para agregar a informação visual de cada pessoa e a relação visual dessa com cada outra presente no vídeo, isto é repetido para todas as pessoas. O *Audio-Visual Relational Context* é utilizado para obter uma representação das características do áudio e cara de cada pessoa no vídeo, para fazer isto estas duas são dadas a uma rede. Que tem ainda na sua fase final o objetivo de baixar a probabilidade das pessoas que dadas as características muito possivelmente a pessoa não está a falar. Os dois resultados do *relational context* são depois concatenados e dados à última fase do modelo, onde é construído o *temporal context*, utilizado para incorporar as informações do tempo nas representações. Depois de obter a representação final dos elementos visuais e áudio-visuais, estas são passadas para a camada onde é feita a previsão.

2.9 Conclusões

É possível concluir que, os modelos embora utilizem métodos diferentes têm uma abordagem muito parecida no sentido em que, começam por extrair as características visuais e do áudio, juntam as representações das várias pessoas para tentar encontrar relações e terminam utilizando o tempo para fazer um último refinamento antes da previsão.

Capítulo

3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Este capítulo descreve as tecnologias e ferramentas utilizadas no desenvolvimento dos programas para o processo de fazer as anotações.

3.2 *YOLOv5*

O *You Only Look Once* (YOLO)v5 é um modelo feito para detetar objetos. Tem este nome porque ao contrário de outras abordagens para esta tarefa, este modelo prevê todas as *bounding boxes* de todas as classes simultaneamente e tem duas grandes vantagens, permitir treino *end-to-end* e grande velocidade nas deteções mantendo a precisão [8]. Para o YOLO fazer a deteção este começa por dividir as imagem em N grelhas, no caso de um objeto aparecer numa dessas grelhas o objetivo dela é detetar o objeto. Cada uma das grelhas prevê várias *bounding boxes* e um valor de confiança, associado a cada uma de que existe ali um objeto. Se não existir objeto naquela área o valor deve ser igual a zero. Estes valores são obtidos durante o treino a partir de *Intersection over Union* (IOU) de previsões de *bounding boxes* com *bounding boxes* de anotações. Aquela que tiver o valor mais elevado é a selecionada. Cada grelha prevê também a que classe pode pertencer o objeto. O modelo é constituido por três componentes principais, o *Backbone*, responsável por extrair características, *Neck* para juntar as características e por fim o *Head* que devolve o tamanho e posição de cada *bounding box*, em conjunto com o valor de confiança e probabilidade de pertencer a uma classe.

3.3 *DeepSort*

O *DeepSort* é uma extensão do método *Simple Online and Realtime Tracking* (SORT) [9], que permite atribuir uma identificação a uma pessoa num vídeo e seguir essa pessoa mantendo a identificação. O *DeepSort* introduz uma rede neuronal convolucional para reduzir a troca de identificações da mesma pessoa ao longo do vídeo. O modelo começa por receber as *bounding boxes* das pessoas no vídeo, neste caso, provenientes do YOLOv5. O primeiro passo utiliza uma abordagem chamada de Kalman *filter*, que ajuda a estimar para cada deteção de um objeto do YOLOv5 a sua posição na *frame* seguinte. Depois, para cada previsão é criado um *track* que contém informação sobre o estado atual do objeto, uma dessas informações é qual foi a última boa previsão do Kalman *filter*, que caso já tenha sido à algum tempo é considerado que a pessoa já saiu da imagem. O resultado são um conjunto de estados para cada objeto, que correspondem a novas *bounding boxes*. É também preciso uma forma de associar novas previsões a deteções do YOLOv5, para isto é utilizada a distância *squared Mahalanobis*, que calcula a distância entre a nova previsão e uma deteção, caso a distância seja superior a valor pré-determinado considera-se que esta previsão não corresponde ao mesmo objeto que está na deteção. A rede neuronal surge para melhorar esta métrica de distância, utilizando as características visuais dos objetos para identificar melhor se uma previsão de uma caixa corresponde a uma deteção.

3.4 *AlphaPose*

O *AlphaPose* utiliza a abordagem *two-step framework*, em que dadas as *bounding boxes* dos corpos das pessoas estima a pose dentro dessas caixas [10]. Neste caso, as *bounding boxes* são dadas pelo YOLOv5, que são passadas ao *Regional Multi-person Pose Estimation* (RMPE) que consiste de três principais componentes. O primeiro componente é o *Symmetric Spatial Transformer Network* (STN) and *Parallel Single-Person Pose Estimator* (SPPE), este pode ser dividido em três passos, o STN que é utilizado para extrair as regiões de maior interesse na imagem, o SPPE utilizado para estimar as poses e o *Spatial Detransformer Network* (SDTN) para colocar a pose estimada na imagem original. O *Parallel SPPE* é utilizado durante o treino para ajudar a treinar o STN para que se foque nas regiões corretas e com mais relevância. O segundo componente é o *Parametric Pose NonMaximum-Suppression* (NMS) que serve para eliminar previsões de poses redundantes. Para fazer isto, é primeiro selecionada a pose que foi determinada com mais confiança e esta é considerada de referência, de seguida são eliminadas poses que são demasiado parecidas.

O último componente é a *Pose-guided Proposals Generator*, utilizado também apenas durante o treino com o intuito de aumentar o conjunto de dados para treino, de forma ao modelo aprender a lidar melhor com imperfeições nas deteções dos corpos das pessoas.

3.5 CVAT

O *Computer Vision Annotation Tool* (CVAT) é uma ferramenta que permite fazer anotações em imagens, no caso deste projeto foi utilizada para anotar o corpo e cara de cada pessoa em cada *frame* de um vídeo através de uma *bounding box* com uma identificação para cada pessoa.

3.6 Conclusões

Estas tecnologias são muito importantes para o desenvolvimento dos programas para fazer as anotações, nomeadamente, iram permitir, como é explicado no próximo capítulo, automatizar o processo de anotações das imagens, tornando-o mais rápido.

Capítulo

4

Anotações

4.1 Introdução

Este capítulo aborda o processo de anotação de vídeos e de áudio utilizado para anotar os vídeos no conjunto de dados. Para ser possível utilizar um conjunto de vídeos para treinar e avaliar um modelo, o conjunto precisa de estar anotado. As anotações consistem de uma *bounding box* à volta da cara das pessoas, um número que identifica a pessoa e uma etiqueta que indica se a pessoa que aparece nela está a falar ou não. Para anotar os vídeos foram utilizados vários programas desenvolvidos em *Python* e *Bash Scripts*. Foi ainda tirado proveito de tecnologias que permitiram anotar as caras das pessoas, em alguns vídeos, de forma automática. Por último, foi preciso desenvolver um programa para converter as anotações para o formato do conjunto de dados do AVA *Active Speaker*.

4.2 Anotações de Vídeo

Nesta secção é descrito o processo para obter as anotações dos corpos e caras das pessoas presentes nos vídeos do conjunto de dados.

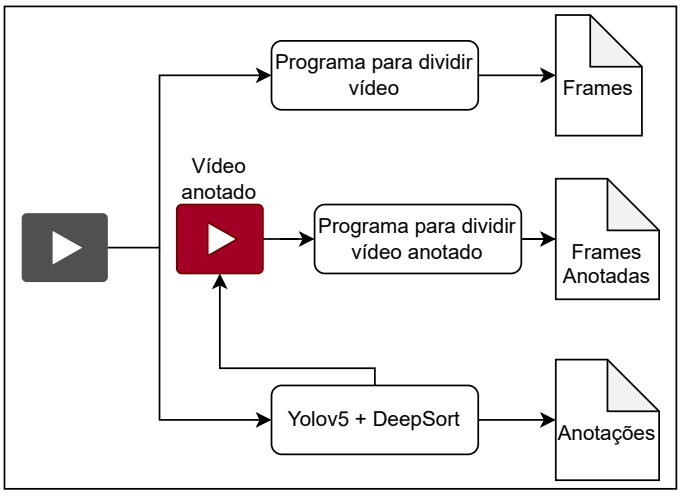

Figura 4.1: *Script* para obter anotações dos corpos.

Para começar o processo de anotar um vídeo o primeiro passo é correr o *script* que é mostrado na figura 4.1 que começa por utilizar o programa para dividir o vídeo em *frames*, estas vão ser utilizadas mais tarde para ser possível observar as anotações. Depois o vídeo é passado ao YOLOv5 e *Deepsort*, o YOLOv5 vai estimar caixas à volta das pessoas em cada *frame* e o *DeepSort* vai atribuir-lhes uma identificação que é mantida ao longo do vídeo. Por fim, é utilizado o programa que divide o vídeo anotado que o YOLOv5 e o *DeepSort* devolveram, em *frames* que são utilizadas para verificar se o *DeepSort* se enganou em alguma identificação ao longo do vídeo.

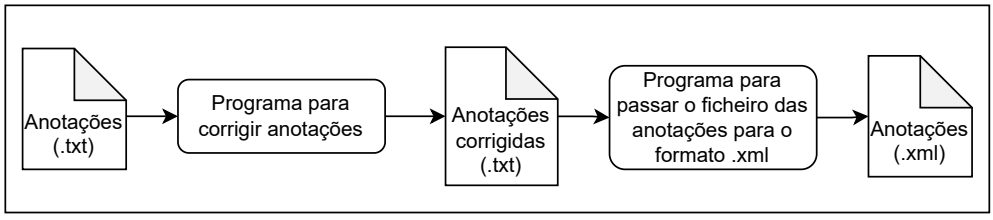

Figura 4.2: *Script* que prepara anotações para poderem ser corrigidas no CVAT.

O segundo *script*, ilustrado na figura 4.2, serve para preparar as anotações dos corpos para poderem ser corrigidas no CVAT. Primeiro é executado o programa que é utilizado para corrigir identificações atribuídas à pessoa errada, no caso de anteriormente ter sido detectado algum erro, e de seguida é criado a partir de um ficheiro de texto com as anotações, devolvido pelo Yolov5

e Deepsort, o ficheiro *Extensible Markup Language* (XML) que é utilizado no CVAT para serem feitas correções. Para se poder fazer estas correções são precisas também as *frames* obtidas no *script* da figura 4.1 a partir do programa que divide o vídeo.

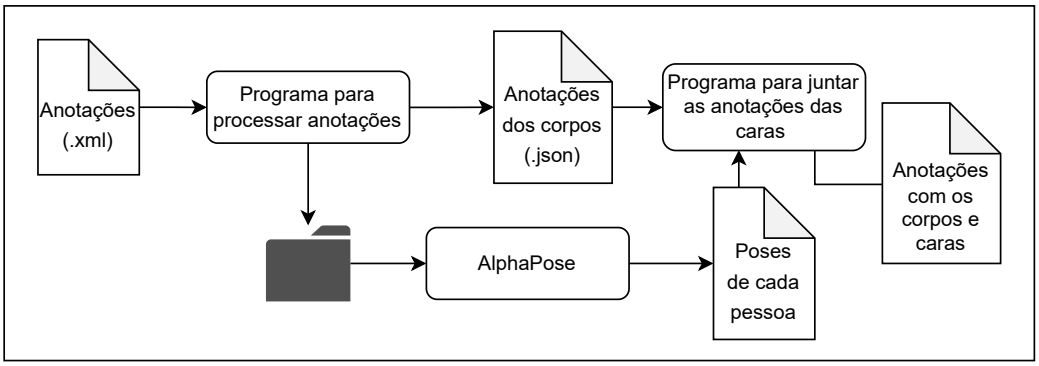

Figura 4.3: *Script* para obter as anotações das caras.

O terceiro *script* vai obter as anotações das caras e juntá-las às anotações dos corpos. Como demonstrado na figura 4.3 as anotações vindas do CVAT são utilizadas, pelo primeiro programa, para recortar o corpo de cada pessoa em cada *frame*, as *frames* utilizadas são as obtidas no *script* da figura 4.1 através do programa que divide o vídeo. Estes recortes são guardados em diferentes imagens dentro de uma pasta, cujo o conteúdo é passado ao *Alphapose* que vai estimar a pose de cada pessoa em cada *frame*. O primeiro programa cria ainda um ficheiro do tipo *JavaScript Object Notation* (JSON) com as anotações. Conhecendo a região da cara das pessoas, é utilizada esta informação pelo segundo programa para criar as anotações das caras e acrescentá-las às anotações dos corpos. Antes de seguir para o próximo passo do processo é preciso correr um outro *script* que converte o ficheiro, com as anotações das caras e corpos, que é do tipo JSON num ficheiro do tipo XML para poderem ser feitas correções no CVAT.

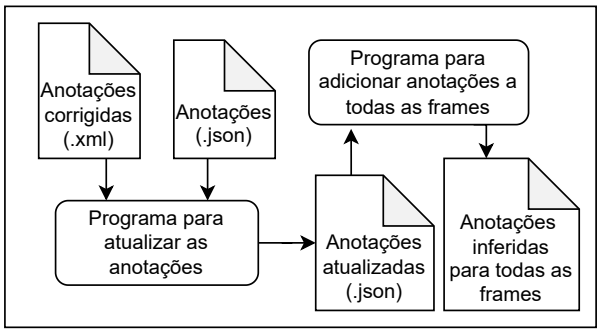

Figura 4.4: *Script* para obter as anotações em todas as *frames* do vídeo.

O *script* representado na figura 4.4 começa por actualizar as anotações no ficheiro do tipo JSON com as anotações vindas do CVAT num ficheiro do tipo XML, isto é feito no programa para atualizar as anotações. Até este momento só foram feitas anotações de cada oito em oito *frames* do vídeo, por isso vai ser utilizado o programa para adicionar anotações a todas as *frames* para com as anotações que se tem actualmente fazer a inferência entre *frames* de modo a ficar com todas as *frames* do vídeo anotadas. Este *script* cria também um ficheiro XML com as anotações actualizadas para todas as *frames*, no caso de ser preciso fazer correcções.

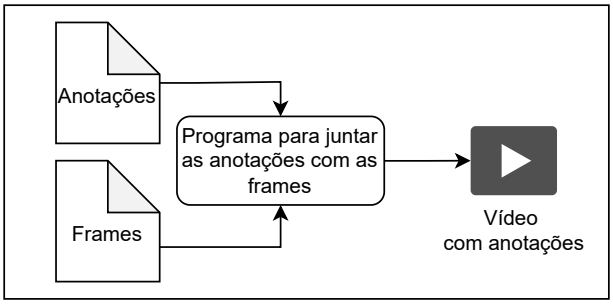

Figura 4.5: *Script* para obter vídeo com anotações.

Para observar as anotações no vídeo este pode ser criado ao utilizar o *script* mostrado na figura 4.5 que junta todas as *frames* com as anotações do ficheiro JSON. Este vídeo é criado com o intuito de ser mais simples a visualização do vídeo com as anotações de modo a ser mais fácil identificar erros. Caso sejam detectados erros é utilizado o ficheiro do tipo XML criado no *script* da figura 4.4 para fazer correcções no CVAT, e é utilizado um *script* para converter o ficheiro vindo do CVAT para um ficheiro do tipo JSON para poder ser criado o vídeo com anotações novamente.

Por fim, quando já não existem erros nas anotações o último *script* a ser executado comprime num ficheiro os ficheiros com as anotações, no formato XML e JSON.

4.3 Anotações de Áudio

Estas anotações servem para indicar qual é a pessoa que está a falar em determinado momento do vídeo.

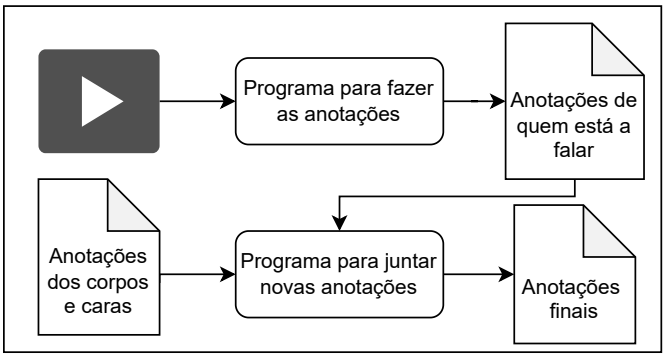

Figura 4.6: *Script* para obter anotações de quem está a falar.

Para fazer estas anotações é utilizado um *script* que é ilustrado na figura 4.6. Para anotar quem está a falar num vídeo o primeiro passo é indicar no programa o número de pessoas presente no vídeo e de seguida associar o número que identifica cada pessoa a uma tecla, que será utilizada durante o processo de anotação para indicar se essa pessoa está a falar. Para iniciar o processo o programa precisa do vídeo, das *frames* do vídeo e das anotações do corpo e da cara. As anotações vão ser utilizadas para desenhar as caixas que delimitam os corpos e as caras das pessoas nas *frames* do vídeo, e o vídeo vai ser utilizado para ser possível saber os *Frames per Second* (FPS) deste e para extrair o áudio, que é reproduzido enquanto são feitas as anotações.

O programa começa por ler o número de FPS, que é utilizado para definir o tempo que o programa espera que uma tecla seja premida, este tempo é dado pela divisão de um número pelo número de FPS, este número pode ser ajustado para fazer esta espera para trocar de *frame* mais longa, utilizando um número maior. Isto pode ser útil quando, por exemplo o vídeo a anotar tem muita gente sucessivamente a falar onde pode ajudar que o vídeo passe mais devagar. No caso de se utilizar este número igual a 1, ou seja o intervalo de tempo igual a $\frac{1}{\text{FPS}}$ o vídeo corre à sua velocidade real. Para se começar a anotar é também preciso o ficheiro de áudio que também é ajustado tendo

em conta o intervalo de tempo entre *frames*, de modo ao áudio e vídeo ficarem sincronizados. Depois as *frames* são mostradas uma a uma, sendo que o programa espera o intervalo de tempo antes de passar para a próxima *frame* e ler de seguida se alguma tecla está a ser pressionada. Esta tecla pode ser uma das utilizadas para representar as pessoas, ou então uma para recuar cinco segundos nas *frames*, no caso de ter acontecido algum erro a anotar. Se foi o primeiro caso o programa utiliza um dicionário onde está a guardar as anotações para saber se estava a ser registado que a pessoa estava a falar. O que significava que foi premida a tecla para indicar que a pessoa deixou de falar, caso contrário o programa começa a registar que a pessoa está a falar. Quando a tecla premida é a utilizada para recuar no vídeo, o programa recua o áudio cinco segundos e volta cinco vezes o número de FPS para trás. Quando se chega à última *frame* é guardado um ficheiro com apenas estas anotações e é utilizado outro programa para juntar os dois tipos de anotação.

No final é possível visualizar um vídeo, que é feito com as *frames* e os dois tipos de anotações, e é também reproduzido ao mesmo tempo o áudio para permitir identificar erros mais facilmente nas anotações.

4.4 Conversão das Anotações para o Formato AVA

Para ser possível passar as anotações para um modelo de *active speaker detection*, de modo a este poder treinar e conseguir fazer previsões, é preciso converter as anotações que se encontram num ficheiro do tipo JSON. Neste ficheiro as anotações são guardadas de forma a que cada *frame* de um vídeo é associada a um número que a identifica e cada anotação que também contém uma identificação contém a da *frame* e da pessoa a que corresponde uma *bounding box*, cuja localização e limites são também guardados, e uma etiqueta que indica se a pessoa está a falar. Esta forma de guardar estas informações é bastante útil por este tipo de ficheiros ser facilmente manipulado em *Python*. E também devido à sua estruturação que permite chegar à anotação que se quer alterar rapidamente, utilizando a identificação da *frame* e da pessoa.

No entanto, os modelos de estado de arte optam por utilizar ficheiros *Comma-separated values*, logo é preciso fazer uma conversão para este formato. Para além do formato, as anotações têm também de ter as informações destas numa certa ordem, de modo a poderem ser lidas mais facilmente. As informações são: um nome que identifica o vídeo a que a anotação se refere; um valor que indica a altura do vídeo em que a *frame* aparece, este valor começa a zero e vai sendo incrementado com um número calculado dividindo a duração do vídeo em segundos pelo número de *frames*, dando assim o intervalo

de tempo passado entre *frames*; quatro coordenadas para indicar a posição e limites da *bounding box*; uma etiqueta que pode ter o valor de *SPEAKING* ou *NOT SPEAKING*, caso a pessoa esteja a falar ou não; uma identificação do nome do vídeo juntamente com o tempo de começo e de final deste, em segundos, e a identificação da pessoa para a qual foi feita a anotação; é ainda dado o número 0 ou 1, para caso a pessoa esteja a falar ou não.

O resultado desta conversão é um ficheiro com o conjunto de todas as anotações dos vídeos do conjunto de dados, e vários ficheiros separados, um para cada vídeo do conjunto de dados.

4.5 Conclusões

Esta parte tem uma grande importância para o trabalho porque para avaliar o modelo são precisos os vídeos anotados para saber se o modelo está a acertar nas suas decisões. E por isto, era muito importante que estas ficassem bem feitas, senão, podia haver situações em que modelo estava certo, no entanto, quando fosse comparar com as anotações, e caso estivessem mal feitas, ia ser considerado que o modelo tinha errado.

Capítulo

5

Avaliação do modelo Active Speakers in Context

5.1 Introdução

Este capítulo aborda a fase do projeto que consistia na avaliação de um modelo de estado de arte para *active speaker detection* utilizando o conjunto de vídeos anotado. A primeira secção fala um pouco sobre o modelo escolhido e razões para a sua escolha, depois segue-se uma descrição dos conjuntos de dados utilizados para a validação e para o treino, e o capítulo termina com uma discussão sobre os resultados obtidos.

5.2 Escolha do modelo

O modelo escolhido para treinar e testar foi o *Active Speakers in Context*. Foi escolhido este modelo porque embora não seja o que tenha os melhores resultados no conjunto de dados AVA *Active Speaker Dataset*, existiram problemas que não permitiram correr os outros modelos propostos. Para além disto, este modelo serviu de base para os outros e, como foi possível concluir do estado de arte, todos os modelos utilizam a mesma estratégia para fazer a tarefa de *active speaker detection*.

5.3 Conjuntos de dados

Foram utilizados dois conjuntos de dados, o AVA *Active Speaker Dataset* e um conjunto de vídeos anotados ao longo do semestre. O primeiro foi utilizado

para treinar e avaliar o modelo, o segundo apenas para avaliação, isto porque não é suficiente o número de vídeos que tem para treinar o modelo.

O AVA *Active Speaker Dataset* consiste de cerca de 38 horas anotadas de vídeos de filmes do *Hollywood*. Contém bastante diversidade na linguagem, no entanto, apresentam cenários em que na maioria das vezes o modelo tem sempre acesso à cara e a um áudio com boa qualidade. Cada pessoa pode ter a etiqueta de *Not Speaking*, *Speaking and Audible* e *Speaking but not Audible*, o último caso para casos em que a pessoa está a falar mas o som não é apanhado pelo microfone.

O conjunto de dados anotado ao longo do semestre contém 4 horas de vídeos retiradas de um conjunto de 30 horas, e é constituído de vídeos que pertencem às 5 categorias seguintes: *React*, *Interview*, *Police*, *Debate* e *Podcast*. Este conjunto foi feito para testar o modelo em cenários em que a qualidade do som e da imagem é diferente, e em vídeos em que a cara da pessoa não está parcialmente ou totalmente visível. Imagens representativas de cada categoria do conjunto de dados podem ser vistas na imagem 5.1.

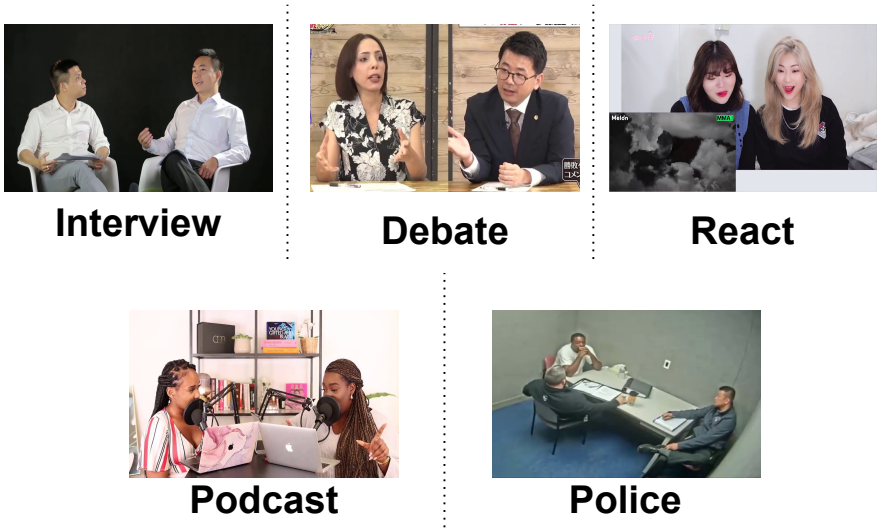

Figura 5.1: Imagens representativas de vídeos do conjunto de dados anotado.

Os vídeos de *Interview* e de *Debate* são os que mais se assemelham aos vídeos no conjunto AVA, isto porque a cara das pessoas está bem visível na imagem e cada pessoa fala de cada vez, e assim, mesmo que alguns fatores como a linguagem variem, os dois elementos mais importantes para o modelo mantém uma boa qualidade.

No conjunto de vídeos *React*, isto já não acontece porque é muito comum uma pessoa falar por cima do áudio do vídeo a que está a reagir. Mesmo man-

tendo uma visão clara da cara da pessoa, o áudio contém também partes em que se ouve alguém a falar, no entanto nenhuma das pessoas no vídeo o está a fazer. Criando assim mais dificuldades ao modelo que tem que lidar com um áudio com interferências. Nos vídeos de *Podcast*, é muito comum as pessoas terem microfones à frente da boca o que faz com que o modelo não tenha acesso às imagens da boca, isto dificulta bastante a previsão do modelo porque foi treinado em cenários em que isto não acontecia e o movimento da um boca é um elemento básico para identificar se alguém está a falar.

Por fim, os vídeos *Police*, que são os que apresentam mais dificuldades para o modelo porque consistem de vídeos filmados de um canto de uma sala em que o vídeo e o áudio é de pouca qualidade e por vezes as pessoas falam por cima das outras. Para além disso, surge um problema que não estava presente nas outras categorias, a cara das pessoas muitas vezes não aparece porque estas estão de costas.

Neste conjunto de dados as pessoas tem apenas a etiqueta *Not Speaking* ou *Speaking and Audible*.

5.4 Discussão dos Resultados Obtidos

O modelo *Active Speakers in Context* foi então treinado durante 100 épocas com o conjunto de dados AVA *Active Speaker Dataset*, sendo que foram escolhidos os parâmetros utilizados na última época do treino para fazer a validação, o modelo foi avaliado no mesmo conjunto de dados e naquele que foi anotado ao longo do semestre. Em relação ao primeiro conjunto foi conseguido obter um valor próximo do reportado pelos autores como é possível ver na tabela 5.1.

Conjuntos de dados	mAP
AVA *Active Speaker Dataset*	80 (87.1)
Anotado	35
Interview	51
Debate	36
Podcast	42
React	16
Police	27

Tabela 5.1: Resultados da avaliação do modelo de estado de arte nos dois conjuntos de dados. O valor entre parênteses indica o valor reportado pelos autores. O conjunto anotado tem o valor para o conjunto completo e individualmente para cada categoria.

Para o conjunto de dados anotado, de modo a perceber melhor que fatores levaram a que eficiência do modelo tenha descido tanto foi feita a avaliação para cada categoria do conjunto. Como é possível ver na tabela 5.1 o conjunto com maior valor *mean Average Precision* (mAP) é o de *Interview*, isto era o esperado uma vez que é aquele que mais se assemelha aos vídeos do conjunto do AVA, no entanto, sendo um conjunto semelhante está ainda bastante distante do valor conseguido nesse, 80 mAP. Acontece que, independentemente de manter a imagem da cara e relativamente bom áudio, existem variáveis como a qualidade destes não ser sempre a melhor, existirem diferentes línguas faladas nos vídeos e sobreposição de fala fazem com que o modelo não consiga replicar o valor obtido no conjunto de dados em que foi treinado. Demonstrando assim que o modelo tem dificuldade em generalizar quando lhe são dados vídeos de tipo diferente dos que lhe foram dados durante o treino, mesmo não divergindo muito destes.

A categoria *Debate* esperava-se mais próximo da *Interview*, porém ficou atrás do *Podcast* revelando que a qualidade do som e não existir sobreposição de vozes são fatores muito importante para o modelo. O facto de que, alterando um pouco a variável do som e mantendo a qualidade da imagem façam com que a capacidade de generalizar do modelo desça, pode significar que o som tem mais influência na decisão do modelo do que a imagem da cara.

Os vídeos de *Podcast* eram dos que apresentavam mais dificuldade a nível de imagem da cara porque muitas vezes as bocas das pessoas não estão visíveis devido aos microfones, o que significa que o modelo perde um dos elementos mais básicos para perceber se uma pessoa está a falar. No entanto, como já dito antes, o facto de a categoria *Debate* ter tido um valor menor pode

indicar que tem mais importância para o modelo a qualidade do som do que ter acesso à imagem da boca.

Na categoria de *React* o resultado obtido foi bastante abaixo do esperado, porém esta categoria é das mais afetadas a nível do áudio porque muitas vezes este não corresponde a nenhuma das pessoas no vídeo a falar. O que dado o resultado da categoria *Debate*, em que a qualidade do áudio também é afetada, pode explicar o valor ter sido tão inferior. O que pode reforça mais a hipótese de o modelo estar bastante dependente da qualidade do áudio. O facto da quantidade de vídeos anotados nesta categoria ser inferior às restantes pode também ter sido um fator que levou a este resultado.

Por fim, como era expectável a categoria *Police* foi aquela em que o modelo teve mais dificuldade, nestes vídeos em que qualidade do vídeo e som é má, e muitas vezes não é possível ver a cara da pessoa, são vídeos em que este modelo não consegue fazer previsão com grande confiança, sendo que algumas vezes vai ser apenas uma questão de sorte acertar se a pessoa estava a falar ou não. Isto, porque o modelo foi pensado e treinado para um conjunto de dados específico que não tem muitos dos problemas apresentados, logo quando são apresentados cenários em que esses problemas estão presentes o modelo deixa de conseguir generalizar, porque as características com que aprende durante o treino não aparecem nestes vídeos.

5.5 Conclusões

Deste capítulo pode concluir-se que depois de treinar o modelo, este teve um desempenho bastante parecido com aquele reportado com os autores, e que com o modelo treinado este não consegue generalizar tão bem em cenários *wild*. O que revela que o modelo não consegue lidar bem com situações reais.

Capítulo

6

Conclusões e Trabalho Futuro

6.1 Conclusões Principais

Deste projeto conclui-se que, após a avaliação do modelo de estado de arte no conjunto anotado, os modelos estão demasiado dependentes de uma boa qualidade de áudio e uma imagem visível da cara. No entanto, cenários onde pode ser útil este tipo de deteção, como por exemplo, câmaras de vigilância, a maioria da vezes não contém essas condições.

6.2 Trabalho Futuro

Para trabalho futuro existiam várias tarefas que permitiam avaliar melhor o desempenho dos modelos de estado de arte de *active speaker detection*, a primeira seria explorar melhor as estatísticas do conjunto de dados o que iria permitir perceber como outros fatores, para além da qualidade do som e da imagem, fazem variar o desempenho do modelo.

Faria também sentido aumentar o conjunto de vídeos anotado até ao ponto de se ter vídeos suficientes para treinar o modelo, isto, porque no conjunto de dados AVA as caras das pessoas estão bem visíveis, com boa qualidade de áudio e sem sobreposição de fala. Isto está distante dos vídeos *wild*, em que muitas vezes o modelo nem sequer tem acesso à imagem da cara. Caso o modelo treinasse em dados que não têm condições quase perfeitas podia significar que este não ficava tão dependente dos elementos a que fica treinado no conjunto AVA.

Seria também interessante testar o conjunto de dados em mais modelos do estado de arte porque embora a abordagem seja muitas vezes parecida os métodos utilizados para extrair características, juntar relações entre cada

pessoa e informação temporal, e fazer a previsão, variam para cada modelo. O que significa que existe o potencial de um dos outros modelos conseguir lidar melhor com imagem e áudio em piores condições.

Bibliografia

[1] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, Caroline Pantofaru. Ava-activespeaker: An audio-visual dataset for active speaker detection. *arXiv:1901.01342*, 2019.

[2] Juan Leon Alcazar, Fabian Caba Heilbron, Long Mai, Federico Perazzi, Joon-Young Lee, Pablo Arbelaez, Bernard Ghanem. Active speakers in context. *arXiv:2005.09812*, 2020.

[3] 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)*, pages 4084–4090, 2020.

[4] Okan Köpüklü, Maja Taseska, Gerhard Rigoll. How to design a three-stage architecture for audio-visual active speaker detection in the wild. *arXiv:2106.03932*, 2021.

[5] Juan León-Alcázar, Fabian Caba Heilbron, Ali Thabet, Bernard Ghanem. Maas: Multi-modal assignation for active speaker detection. *arXiv:2101.03682*, 2021.

[6] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. *arXiv:2107.06592*, 2021.

[7] Yuanhang Zhang, Susan Liang, Shuang Yang, Xiao Liu, Zhongqin Wu, Shiguang Shan, Xilin Chen. Unicon: Unified context network for robust active speaker detection. *arXiv:2108.02607*, 2021.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv:1506.02640*, 2015.

[9] Nicolai Wojke, Alex Bewley, Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *arXiv:1703.07402*, 2017.

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, Cewu Lu. Rmpe: Regional multi-person pose estimation. *arXiv:1612.00137*, 2016.