Kaio: Detector de Expressões Faciais e Classificador de Emoções

1. Introdução

Vivemos em um mundo cada vez mais conectado e imerso em ambientes repletos de dispositivos computadorizados. Tal fenômeno proporcionou o surgimento de estudos em temas como Computação Ubíqua e Pervasiva [Satyanarayanan 2001] e Internet das Coisas [Gubbi et al. 2013]. Desenvolvimentos nestes temas trazem como consequência a necessidade cada vez maior de interação entre o ser humano com máquinas e dispositivos eletrônicos. Novas formas de interação como uso de gestos [Mitra and Acharya 2007] e comandos de voz [Cohen et al. 2004] podem trazer mais agilidade e conforto para que o ser humano realize suas tarefas. Mas tem-se a preocupação em evitar que tais atividades sejam muito frias e monótonas.

O uso de interfaces homem-máquina que levem em consideração as emoções humanas vem sendo alvo estudos recentes. Por exemplo, [Fragopanagos and Taylor 2005], levantam a necessidade de que interações mais naturais com dispositivos sejam baseadas em forma de reconhecimento de emoções, permitindo a esses dispositivos se adaptarem e otimizarem o processo interativo. Assim, propõem o uso de técnicas de inteligência artificial sobre imagens e vozes para a detecção das emoções do usuário. No entanto, não discutem métodos objetivos de como tais técnicas poderiam compor aplicações práticas.

Neste artigo, propomos uma arquitetura baseada em visão computacional para detecção de emoção a partir de expressões faciais e apresentamos um estudo de caso focado na interação com personagens digitais. A arquitetura proposta visa na avaliação de métodos adequados de aprendizado de máquina e formas de interação com diferentes dispositivos. A aplicação em personagens digitais visa servir de modelo, embora também tenha a motivação de ser aplicado em situações de aprendizado de crianças, pessoas com necessidade especiais ou simples entretenimento.

O restante do artigo está estruturado da seguinte forma. Nesta seção apresentamos o problema e as métricas. A seção 2 apresenta uma análise detalhada sobre os dados. A seção 3 apresenta a arquitetura proposta e a seção 4 discute os resultados obtidos da experimentação. Finalmente, a seção 5 conclui o trabalho e apresenta trabalhos futuros.

1.1. Problema

O uso de ferramentas que levem em consideração as emoções humanas ainda são alvo de estudo em muitas pesquisas, pois problemas como este focam em encontrar uma forma de tornar mais atraente e natural a interação homem-máquina.

O objetivo deste trabalho é construir um sistema inteligente que leve em consideração as emoções humanas por meio da detecção de expressões faciais para interagir com um personagem digital.

Para tal será necessário construir um modelo de aprendizagem que classifique emoções baseado em métodos automáticos de aprendizagem de máquina. Portanto, será necessário o uso de técnicas destinadas a classificação multiclasse. E para simplificar esta tarefa, utilizaremos entradas de texto produzidas por meio de um detector facial a

parte, feito com a biblioteca *Android Mobile Vision - AMV* [Google 2017], em um módulo separado do classificador, para fornecer as características necessárias da detecção.

Uma vez definido o problema, para cada sequência de detecção produzida teremos 4 (quatro) características e 1 (uma) classe alvo sobre cada usuário. Esta detecção será feita pela biblioteca AMV em um aplicativo Android.

Para ficar claro o funcionamento desta detecção, vejamos uma sequência de expressões faciais detectada pela biblioteca AMV para um usuário com aparência de tristeza, conforme a Tabela 1:

Table 1. Sequência de expressões de tristeza

user	rate blink left	rate blink right	rate smile or not	feel
1	0.99	0.99	0.01	0
1	0.99	0.99	0.01	0
1	0.92	0.87	0.02	0
1	0.92	0.87	0.02	0
1	0.92	0.87	0.02	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.02	0

Neste exemplo a pessoa teve a liberdade de piscar qualquer um dos olhos, sorrir ou apenas abrir a boca, e todas estas ações serão interpretadas pelo detector como uma sequência de expressões faciais que serão compreendidas como emoções, e consequentemente, convertida em probabilidades, significando:

- Probabilidade de uma pessoa está sorrindo (*rate smile or not*)
- Probabilidade de uma pessoa estar com olhos abertos (*rate blink left or right*).

As probabilidades são valores entre 0.0 e 1.0 (inclusive) através dos campos **rate blink left, rate blink right** e **rate smile or not**, já as emoções são fornecidas pelo campo **feel** que corresponde ao nosso alvo para o modelo de aprendizagem, variando entre 3 tipos, tais como: 0 - Tristeza; 1 - Raiva; e 2 - Felicidade.

Portanto, o objetivo do nosso classificador dado uma sequência de expressões faciais é classificar uma emoção em tristeza, raiva ou felicidade.

1.2. Métricas

No contexto de Recuperação de Informação (RI) e Inteligência Artificial (AI), o desempenho de um sistema inteligente para classificação de emoções pode ser mensurado por meio de métricas, tais como: acurácia, precisão, matriz de confusão e validação cruzada, conforme afirmam [Ikonomakis et al. 2005], [MIKAMI et al. 2009] e [Freire et al. 2009].

Acurácia: Segundo [Sabbagh 2011], mensura o total de classificações corretas dividido pelo total de exemplos. Calculada conforme a Figura 1.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FN + FP + TN)}$$

Figure 1. Acurácia

Precisão: Segundo [Freire et al. 2009], mensura o quanto que foi exato a resposta, em uma proporção de classificações relevantes. Calculada conforme a Figura 2.

$$\text{Positive Predictive Value or Precision} = \frac{TP}{(TP + FP)}$$

Figure 2. Precisão

Matriz de confusão: Segundo [Freire et al. 2009], a medida oferece uma efetiva avaliação do modelo de classificação considerando o número de classificações corretas e as classificações estimadas para cada classe em um conjunto de dados. Calculada por meio da Figura 3.

	Predicted	
	Positive	Negative
Actual True	**TP**	**FN**
Actual False	**FP**	**TN**

Figure 3. Matriz de confusão

Validação Cruzada: Segundo [MIKAMI et al. 2009] corresponde ao processo de divisão dos dados coletados em uma parte para treinamento e outra parte para testes. Segundo o autor, é um processo importante para analisar os resultados alcançados pelo modelo de aprendizagem, testando o sistema com dados reais diferente dos dados que ele foi treinamento.

2. Análise

Nesta seção é descrito a origem dos dados, bem como as características e informações da base de detecções faciais que serão utilizadas pelo classificador de emoções.

2.1. Dados

A base de treinamento contendo as expressões faciais de usuários foi coletada em uma fase anterior ao uso do sistema, que ocorreu em 2 dias, com 12 pessoas. O treinamento capturou um total de 360 expressões faciais por usuário, divididas em 60 expressões para 1 emoção (tristeza, raiva ou felicidade). Entre as pessoas envolvidas, 2 foram mulheres, 10 homens, considerando pessoas com barba, sem barba, usando óculos e sem óculos.

Conforme foi explicado, a base de treinamento foi coletada a partir da detecção facial feita pela biblioteca AMV que foi executada em um aplicativo Android e utilizada por 12 pessoas em uma fase anterior ao uso do sistema.

Dentro da base de treinamento encontramos 3 (três) tipos de emoções que variam em 0 - tristeza, 1 - raiva e 2 - felicidade. Vejamos alguns trechos recortados da base de treinamento nas Tabelas 2, 3 e 4.

Table 2. Sequência de expressões do tipo 0 - tristeza

user	rate blink left	rate blink right	rate smile or not	feel
1	0.99	0.99	0.01	0
1	0.99	0.99	0.01	0
1	0.92	0.87	0.02	0
1	0.92	0.87	0.02	0
1	0.92	0.87	0.02	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.08	0
1	0.99	0.99	0.02	0

Para esta sequência da Tabela 2 é possível ver que o usuário manteve os olhos abertos e não estava sorrindo, isso fica claro nos valores para '**rate smile or not**' abaixo de 0.1 e '**rate blink left**' e '**rate blink right**' acima de 0.8.

Table 3. Sequência de expressões do tipo 1 - raiva

user	rate blink left	rate blink right	rate smile or not	feel
1	0.99	0.51	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
1	0.81	0.76	0.01	1
1	0.81	0.76	0.01	1
1	0.81	0.76	0.01	1
1	0.81	0.76	0.01	1

Para esta sequência da Tabela 3 observamos um comportamento bem similar a Tabela 2, com a diferença de que o usuário neste estado manteve os olhos um pouco mais fechado do que o estado de tristeza. Observando os valores para '**rate blink left**' e '**rate blink right**' abaixo de 0.8 e próximos de 0.5, isso revela que olhos estão meio abertos.

Table 4. Sequência de expressões do tipo 2 - felicidade

user	rate blink left	rate blink right	rate smile or not	feel
1	0.99	1.0	0.48	2
1	0.99	1.0	0.48	2
1	0.99	1.0	0.48	2
1	0.99	1.0	0.48	2
1	1.0	0.99	0.57	2
1	1.0	0.99	0.57	2
1	1.0	0.99	0.57	2
1	1.0	0.99	0.42	2
1	1.0	0.99	0.42	2
1	1.0	0.99	0.42	2
1	1.0	0.99	0.31	2

Para esta sequência da Tabela 4 observamos um comportamento totalmente diferente dos estados anteriores, aqui temos o usuário com olhos bem abertos demonstrando sorrisos. Este comportamento fica claro nos valores para '**rate smile or not**' próximos de 0.5 demonstrando a presença de sorrisos, além dos valores de '**rate blink left**' e '**rate blink right**' em 0.99 e 1.0 demonstrando confirmação de olhos super abertos.

O modelo de aprendizagem contém **4264 expressões faciais**, que foram salvas no arquivo **detect.csv**, contemplando os 12 usuários e as 3 emoções. Esse treinamento serviu como subsídio para o sistema ser capaz de reconhecer as emoções. A base de treinamento com as características e a classe alvo salvas no arquivo **detect.csv** foram organizadas da seguinte maneira:

Características:

- *user*: Identificação do usuário que executou o treino;
- r*ate_blink_left*: Probabilidade do olho esquerdo está aberto (0.0 a 1.0);
- *rate_blink_right*: Probabilidade do olho direito está aberto (0.0 a 1.0);
- *rate_smile_or_not*: Probabilidade da pessoa está sorrindo (0.0 a 1.0);

Alvo:

- *feel*: Emoção do usuário (0-tristeza , 1-raiva, 2-felicidade).

2.2. Algoritmo de aprendizagem

Segundo [Calvo and D'Mello 2010], [Forsyth and Ponce 2011] e [Pradhan 2013], as principais técnicas empregadas para problemas de classificação geralmente são Modelos Estatísticos, Redes Neurais, Modelos Genéticos e Árvores de Decisão. Algumas técnicas como os modelos Estatísticas se destacam em problemas textuais, como a detecção de spam em e-mails, parecido com o nosso problema em estudo.

Neste trabalho exploramos praticamente quase todas as principais técnicas disponíveis pela biblioteca Scikit-learn [Pedregosa et al. 2011] - a biblioteca destinada a exploração e análise de dados aplicando aprendizagem de máquina. Entre as técnicas disponíveis na documentação utilizaremos:

- AdaBoostClassifier
- DecisionTreeClassifier
- ExtraTrees
- GaussianNB
- KNeighborsClassifier
- Linear SVC
- MLPClassifier
- MultinomialNB
- OneVsOne
- OneVsRest
- QuadraticDiscriminantAnalysis
- RBF SVM
- SVC with Kernel Linear

Para simplificar explicaremos à alto nível sobre o funcionamento de três tipos de algoritmos diferentes, são eles o Naive Bayes, Extra Tree e K-Nearest Neighbors.

Naive Bayes: Segundo [Wikipedia 2017b] é um algoritmo da família dos classificadores probabilísticos com base no teorema de Bayes, os fortes (ingênuos) pressupostos de independência entre os recursos. São algoritmos escaláveis e com tempo linear. Considerado um dos mais simples de implementar e com boa generalização para muitos padrões.

Extra Tree: Segundo [Johnson 2017] o uso de *Extra Tree* pode ser muito mais interessante do que o uso de uma simples *Decision Tree*, por conta desta técnica combinar diferentes classificadores (árvores de decisão) sem utilizar ***bagging*** construindo divisões aleatórias para gerar árvores diferentes.

K-Nearest Neighbors: Segundo [Wikipedia 2017a] ***KNeighbors*** ou **KNN**, é um algoritmo que recebe como entrada um inteiro K para definir quantos exemplos de treinamentos mais próximos deverão ser encontrados. O resultado é uma associação de classe, elencando um objeto por meio do voto maioritário de seus vizinhos. É um tipo de aprendizagem baseada em instâncias, ou aprendizado preguiçoso, onde a função é apenas aproximada localmente e toda a computação é adiada até a classificação. O algoritmo k-NN é um dos algoritmos de aprendizado de máquina mais simples de todos.

2.3. Benchmark model

Construiu-se um algoritmo base capaz de predizer sempre a mesma emoção, de maneira que, para toda sequência de características recebida o algoritmo prediz sempre a mesma emoção. O objetivo deste benchmark foi construir um algoritmo ”burro” para servir de base para a comparação aos algoritmos inteligentes que serão avaliados.

No final dos experimentos na seção 3.4 será possível perceber que a taxa de acertos do algoritmo base ficou em **36,77%**, muito abaixo dos algoritmos inteligentes, algo que era esperado, pois considerando que o funcionamento do algoritmo era simples e o mesmo só conhecia a mesma saída para qualquer sequência de detecção facial, o resultado seria sempre o mesmo.

3. Kaio: Detector de Expressões Faciais para Classificar Emoções

Esta seção apresenta uma visão geral da ferramenta proposta para detectar expressões faciais e classificá-las em emoções, descrevendo a sua arquitetura, pré-processamento, implementação e processo de classificação.

3.1. Descrição da Ferramenta

A solução proposta consiste na arquitetura de uma ferramenta capaz de possibilitar uma nova forma de interatividade com personagens digitais. Diante disso, utilizando técnicas de aprendizagem de máquina e processamento de imagens, a ferramenta deverá classificar emoções por meio da detecção de expressões faciais. Assim, será possível acessar as informações de interação classificadas em níveis de emoção do usuário, a princípio dividida em: **Tristeza**, **Raiva** e **Felicidade**.

3.2. Arquitetura da Ferramenta

A arquitetura proposta foi distribuída em 3 (três) componentes, uma arquitetura modular com o objetivo de facilitar o desenvolvimento e garantir um baixo acoplamento de cada componente, conforme a Figura 4.

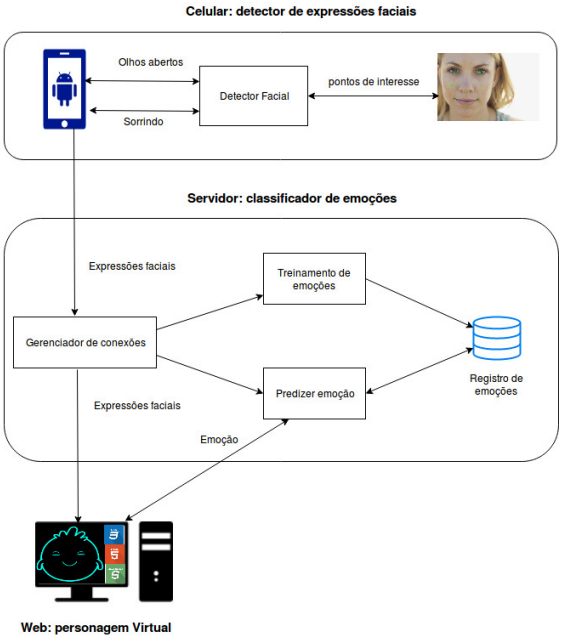

Figure 4. Arquitetura da Ferramenta

3.2.1. Celular: Detector de Expressões Faciais

Este componente é responsável por realizar a detecção das expressões faciais de um usuário por meio do uso de um celular com câmera e sistema operacional *Android*. O componente monitora pontos de interesses, tais como: olhos e boca; por meio do uso da biblioteca AMV. Esta biblioteca fornece informações textuais sobre o quanto que um usuário está sorrindo (*smilling*) e as chances de estar com olhos abertos (*eyes open*). O arquivo **detect.csv**, está organizado conforme a Tabela 5.

Table 5. Sequência de expressões faciais

user	rate blink left	rate blink right	rate smile or not	feel
1	0.99	0.51	0.01	1
1	0.99	0.79	0.01	1
1	0.99	0.79	0.01	1
...	...	...	...	...
1	0.81	0.76	0.01	1

Essa biblioteca AMV possui um conjunto de recursos para detecção facial e classificação de características facial, permitindo aplicações derivadas, saber a partir de um vídeo ou imagem detectar rostos e pessoas, extraindo características do tipo: a pessoa está sorrindo? A pessoa está de olhos abertos? A Figura 5 demonstra os pontos de detecção da biblioteca.

Figure 5. Pontos de detecção da biblioteca AMV

A partir deste componente de detecção foi possível construir a base de treinamento. Para mais detalhes sobre a biblioteca AMV, a documentação oficial está disponível na parte de desenvolvedores no site do [Google 2017].

3.2.2. Servidor: Classificador de Emoções

O principal componente neste trabalho é responsável por classificar emoções por meio da detecção de expressões faciais. Neste encontra-se os módulos para treinamento e predição, além do gerenciador de conexões.

- **Módulo de Treinamento**: Na Figura 4 o módulo ”Treinamento de emoções” corresponde a fase que o sistema precisa aprender por um determinado tempo expressões faciais de usuários. Cada usuário ao submeter-se a detecção de expressões faciais, fornece para o sistema informações do tipo: taxa de olho esquerdo aberto; taxa de olho direito aberto; e taxa de sorriso. Estas informações são primordiais para a construção do modelo neste trabalho, e servirão como referência para classificar as emoções, entretanto, é importante frisar, que o treinamento não ocorre durante o uso do sistema, e sim em uma fase anterior, conforme será descrito na seção 4;
- **Módulo de Predição**: Na Figura 4 o módulo “Predizer emoção” corresponde a fase que o sistema já possui uma base de conhecimento com o registro de emoções, já passou pela fase de treinamento, e conhece algumas expressões faciais. Desta maneira, emprega-se um modelo de aprendizagem para ser testado e avaliado em cima da base de conhecimento, preparando para predições. No final, as emoções classificadas irão variar em três níveis, indicando a emoção do usuário em: “Tristeza”, “Raiva” e “Felicidade”.

- **Módulo Gerenciador de Conexões**: Na Figura 4 o módulo ”Gerenciador de Conexões” é o responsável por delegar as operações solicitadas ao servidor que poderá optar por ativar dois módulos: treinamento ou predição.

3.2.3. Web: Personagem Virtual

Este componente é responsável por reproduzir a nova forma de interação proposta neste trabalho, e foi desenvolvido com o objetivo de demonstrar as possibilidades de interação. Ele recebe o processamento da detecção, e executa as animações de um personagem digital, denominado ”Kaio”, que de acordo com o estado do usuário, assume os estados de piscar e sorrir, imitando o usuário. No final da interação, este componente prediz qual estado emocional o usuário se encontra, classificado nos três níveis citados anteriormente.

3.3. Pré-processamento de dados

Para as expressões faciais contidas no arquivo **detect.csv**, existe um total de **4264 amostras** formatadas em CSV, organizada em características: *user; rate blink left; rate blink right; rate smile or not*; e uma classe alvo *feel*.

Nesta amostra possuímos o reconhecimento facial das 3 emoções sinalizadas pela classe alvo *feel*.

3.4. Implementação

Para escolher o melhor modelo de aprendizagem para o sistema, foi elaborado um comparativo empírico baseado na acurácia entre as principais técnicas de aprendizagem de máquina destinadas a classificação, conforme a Figura 6.

Observou-se que a técnica de melhor desempenho dentre todas, ou seja, o algoritmo que melhor se adaptou, foi o **DecisionTreeClassifier**, conforme os resultados. E todos os algoritmos inteligentes foram melhor que o algoritmo base, o que já era esperado, considerando o simples funcionamento do algoritmo.

Decision Tree: Segundo [Du and Zhan 2002] se destaca em relação ao fácil nível de interpretação, uma classificação explícita, além de baixo custo computacional gerado pela técnica. Outro aspecto interessante, segundo [Kołakowska 2013], são os desempenhos desta técnica, pois dependendo das configurações, o algoritmo pode melhorar sem a necessidade de mais informações.

Para esta versão implementamos a DecisionTree com as configurações padrões disponíveis no Scikit-learn, com exceção do parâmetro random state que foi setado um valor fixo.

Algoritmo	Taxa de acerto (%)
DecisionTreeClassifier	84.54%
ExtraTrees	74.71%
KNeighborsClassifier	72.21%
AdaBoostClassifier	63.28%
RBF SVM	62.60%
OneVsOne	61.30%
Linear SVC	60.96%
OneVsRest	60.96%
QuadraticDiscriminantAnalysis	59.81%
GaussianNB	59.47%
MLPClassifier	59.34%
MultinomialNB	57.21%
SVC with Kernel Linear	53.66%
Algoritmo base	36.77%

Figure 6. Ranking dos algoritmos com maior taxa de acerto

3.5. Refinamento

Para refinar o modelo utilizamos GridSearch e como resultado encontramos uma pequena melhoria do algoritmo em torno de 0,5 pontos percentuais de melhoria.

Antes de aplicar o GridSearch, implementamos uma Decision Tree com os parâmetros padrões, setando o random state em 0, da seguinte maneira:

- class weight=None
- criterion='gini'
- max depth=None,
- max features=None
- max leaf nodes=None,
- min impurity split=1e-07
- min samples leaf=1
- min samples split=2
- min weight fraction leaf=0.0
- presort=False
- random state=0
- splitter='best'

Diante disso, esperava-se que alterando os parâmetros de peso aplicado pelo **class weight** e o critério de divisão da árvore por meio do **criterion**, o mínimo de divisão de amostras por meio do parâmetro **min samples split**, o número aleatório gerado pelo **random state** e o parâmetro **presort** para reorganizar a árvore, fossem bons candidatos para otimizar o modelo e gerar melhores resultados ao algoritmo. Portanto, definido os parâmetros, aplicamos o GridSearch e a melhor configuração para a Decision Tree encontrada foi:

- class weight=None
- criterion='gini'
- max depth=None,
- max features=1
- max leaf nodes=None,
- min impurity split=1e-07
- min samples leaf=1
- min samples split=2
- min weight fraction leaf=0.0
- presort=False
- random state=0
- splitter='best'

O resultado mostrou que praticamente a melhor configuração do algoritmo é a configuração padrão utilizada, com a diferença do **max features** que foi setado em 1 e revelando uma melhoria em 0,5 pontos percentuais. Este parâmetro nas configurações iniciais é setado como o total de características disponíveis, no nosso caso era setado como 4, quando o algoritmo era implementado com as configurações padrões, e automaticamente encontrava o número 4 como características disponíveis, devido ao modelo em estudo.

4. Experimentos e Resultados

Esta seção descreve os dados que foram analisados e os experimentos para avaliar o sistema de classificação das emoções.

4.1. Avaliação

A avaliação do sistema ocorreu em 3 dias, com 10 pessoas diferente das pessoas voluntárias do treinamento. Nesta fase, foram consideradas as seguintes regras:

- Avaliar com 10 pessoas voluntárias;
- Avaliar 3 vezes com a mesma pessoa;
- Permitir que a mesma pessoa escolha a ordem que desejar treinar, sem exceder as 3 tentativas por pessoa;
- Permitir que a pessoa voluntária anote a emoção que ela realizou e qual o sistema classificou;
- Ter 2 pessoas avaliadoras para julgar a emoção que a pessoa voluntária executou.

No final da avaliação, uma base de 30 observações foi adquirida, que serviu de referência para análise e os resultados alcançados neste trabalho. Para cada observação, as pessoas envolvidas e os avaliadores julgaram uma emoção, e a pessoa voluntária observou também a emoção que o sistema realmente classificou. As emoções foram classificadas em: 0 - Tristeza; 1 - Raiva; e 2 - Felicidade.

4.2. Análise e Resultados

Nesta fase foram empregadas as métricas descritas, analisando o desempenho do sistema.

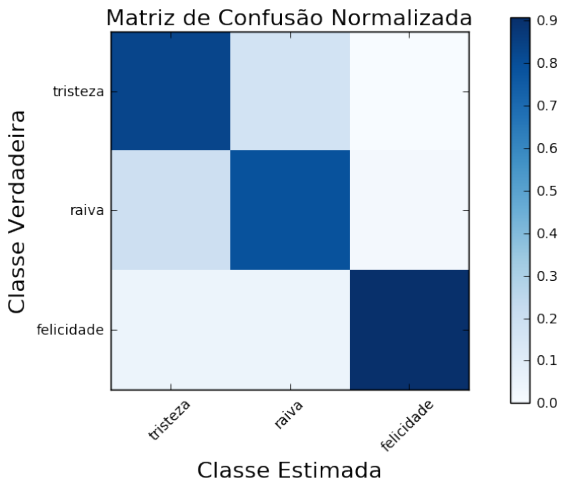

Figure 7. Matriz de Confusão do Sistema

4.2.1. Validação Cruzada

Dividiu-se 90% das 4264 expressões faciais para treino e 10% para testes, então com uma matriz de confusão foi possível verificar o desempenho do sistema, conforme a Figura 7.

De forma geral, observa-se que o sistema atingiu uma alta precisão para classificar as emoções, pois obteve mais acertos do que erros. A emoção denominada como felicidade, o sistema parece não confundir tanto com as outras emoções, entretanto, para tristeza e raiva o sistema parece se confundir algumas vezes entre as duas emoções.

4.2.2. Comparativo com Avaliação Humana

Nesta fase de avaliação, comparamos a precisão do sistema versus a observação humana feita por dois avaliadores, baseado nos testes feitos pelas 10 pessoas, durante a fase de avaliação. Conforme a demonstra Figura 8.

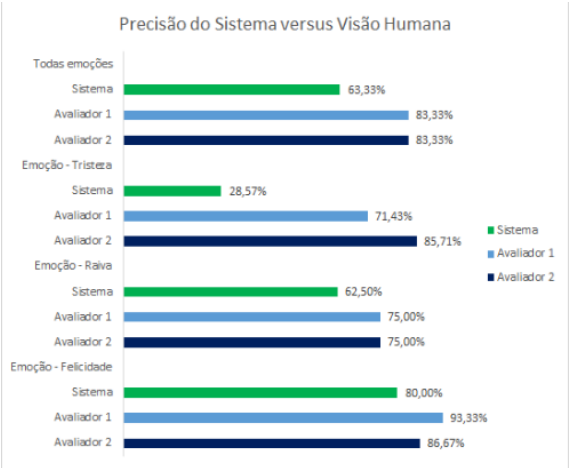

Figure 8. Precisão do Sistema versus Visão Humana

As três primeiras barras correspondem às precisões das avaliações de todas as emoções, classificadas pelo sistema, avaliador 1 e avaliador 2, em relação aos testes feitos pelas pessoas voluntárias durante a avaliação. Pode-se observar que a precisão dos avaliadores humanos em relação à emoção detectada pelos avaliadores foi de 83%, o que mostra um ruído natural na comunicação entre humanos. Já o sistema proposto acertou 63%, 20 pontos percentuais a menos que os avaliadores humanos. Isso se deve à menos características do sistema na detecção de pontos de expressão da face.

As outras barras na Figura 8, demonstram o desempenho do sistema versus a visão humana para cada emoção de forma individual. É possível observar que os acertos do sistema em relação aos avaliadores se manteve em 20 pontos percentuais a menos para a emoção de Raiva, no caso da Felicidade, os acertos ficaram entre 6 e 13 pontos percentuais a menos em relação aos avaliadores, relevando a melhor classificação do sistema. Entretanto, para Tristeza, os acertos não foram maiores que os erros, revelando que o sistema mostrou dificuldade para classificar corretamente esta emoção.

5. Conclusão

O trabalho permitiu avaliar e apresentar um estudo de caso focado na interação com personagens digitais para classificar a emoção do usuário por meio da detecção de expressões faciais. Os resultados mostram que mesmo seres humanos apresentam erros ao interpretar emoções de outros seres humanos, mas que métodos automáticos, como o proposto neste trabalho, podem ser úteis em alguns cenários.

Para algumas emoções como raiva e tristeza, o sistema não teve um desempenho tão alto quanto a classificação da felicidade, devido aos poucos atributos fornecidos pela biblioteca *Android Mobile Vision* para detecção facial. No entanto, a possibilidade de acertar a emoção felicidade com precisão de 80% já pode ser muito útil para aplicações de entretenimento infantil como proposto. A biblioteca oferece apenas 3 pontos de interesse - olhos (esquerdo e direito) e boca, talvez se fosse possível ter outros pontos de detecção na biblioteca como a sobrancelha, poderia permitir diferenciar a Raiva de Tristeza por meio de mais características, mas esta proposta fica para trabalhos futuros.

Outra proposta interessante, seria investigar em trabalhos futuros o uso de outros algoritmos como Redes Neurais, Modelos Genéticos e Random Forest, destacados no estado da arte a fim de encontrar resultados melhores.

Todavia, espera-se que o trabalho possa servir como modelo para interação entre diferentes dispositivos e principalmente para personagens digitais. E se for possível aplicar a ferramenta no entretenimento ou até mesmo como ferramenta educacional, será um ganho a mais para a pesquisa.

References

Calvo, R. A. and D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37.

Cohen, M. H., Cohen, M. H., Giangola, J. P., and Balogh, J. (2004). *Voice user interface design*. Addison-Wesley Professional.

Du, W. and Zhan, Z. (2002). Building decision tree classifier on private data. In *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14*, CRPIT '14, pages 1–8, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Forsyth, D. and Ponce, J. (2011). *Computer vision: a modern approach*. Upper Saddle River, NJ; London: Prentice Hall.

Fragopanagos, N. and Taylor, J. G. (2005). Emotion recognition in human–computer interaction. *Neural Networks*, 18(4):389–405.

Freire, P. P., de Oliveira Lombardi, L., Ciferri, R. R., Thiago Alexandre Salgueiro Pardo, C. D. d. A. C., and Vieira, M. T. P. (2009). Relatório técnico: Métricas de avaliação. projeto: Um ambiente para análise de dados da doença anemia falciforme.

Google (2017). Detect Faces. URL: https://developers.google.com/vision/ Access in 2017–03–22.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

Ikonomakis, M., Kotsiantis, S., and Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974.

Johnson, R. (2017). Decision Tree Ensembles. URL: https://www3.nd.edu/ Access in 2017–08–27.

Kołakowska, A. (2013). A review of emotion recognition methods based on keystroke dynamics and mouse movements. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, pages 548–555. IEEE.

MIKAMI, R., SANTOS, L., VENDRAMIN, A., and KAESTNER, C. (2009). Procedimentos de validação cruzada em mineração de dados para ambiente de computação paralela. *Artigo do Departamento Acadêmico de Informática Universidade Tecnológica Federal do Paraná, Curitiba, Brasil*.

Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pradhan, B. (2013). A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using gis. *Computers & Geosciences*, 51:350–365.

Sabbagh, R. (2011). Precisão e acurácia em estimativas ágeis. URL: https://www.infoq.com/br/articles/precisao-acuracia-agile/ Access in 2017–03–22.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal communications*, 8(4):10–17.

Wikipedia (2017a). k-nearest neighbors algorithm, note = URL: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm Access in 2017–08–27.

Wikipedia (2017b). Naive Bayes classifier, note = URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier Access in 2017–09–08.