



TRABALHO DE CONCLUSÃO DE CURSO

**APLICAÇÃO DE REDES NEURAIS CONVOLUCIONAIS  
NA IDENTIFICAÇÃO DE TATUADORES**

André Henrique Macedo da Costa

Brasília, dezembro de 2018

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA



UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

**APLICAÇÃO DE REDES NEURAIS CONVOLUCIONAIS  
NA IDENTIFICAÇÃO DE TATUADORES**

**André Henrique Macedo da Costa**

*Relatório submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Mylène Christine Queiroz de Farias, \_\_\_\_\_

ENE/UnB  
*Orientador*

Prof. Daniel Guerreiro e Silva, ENE/UnB \_\_\_\_\_

*Co-orientador*

Prof. Alexandre Ricardo Soares Romariz, \_\_\_\_\_

ENE/UnB  
*Examinador interno*

Prof. Alexandre Zaghetto, CIC/UnB \_\_\_\_\_

*Examinador interno*

## **Dedicatória**

*Aos meus pais que são as pessoas que mais me deram suporte nessa jornada.*

*André Henrique Macedo da Costa*

## Agradecimentos

*Existem muitas pessoas que eu gostaria de agradecer e tenho certeza que meus sentimentos não caberiam em apenas uma página. Gostaria de agradecer a minha família por todo o amor, atenção e carinho que recebi. Gostaria de agradecer aos meus avós maternos, Antônio e Teresa, por me mostrarem que é possível vencer as mais duras provas. Vocês foram um exemplo de trabalho duro e amor sem igual. Em especial, gostaria de agradecer aos meus pais, Pedro e Teresa, que provavelmente são as pessoas mais responsáveis pelas vitórias que tive na vida. Obrigado pela paciência e pelos ensinamentos constantes apesar dos meus esquecimentos. Vocês são responsáveis pelas melhores memórias que eu consigo me lembrar. Obrigado por terem me levado para morar na Inglaterra, foi a melhor experiência que tive na vida. Agradeço a minha irmã, Camila, pelo amor e por me antagonizar durante toda a minha vida. Nossas diferenças nos tornaram pessoas melhores. Você foi sempre uma pessoa com quem me comparei. Nessas comparações pude aprender muito.*

*Agradeço aos amigos que se mantiveram ao meu lado durante todos esses anos. A amizade de vocês foi sempre muito importante para mim. Vocês me ensinaram muitas coisas ao longo desses anos. Obrigado Paula pelos almoços desse semestre. Nossos encontros foram uma terapia nesse semestre muito movimentado. Não poderia me esquecer dos amigos que fiz no curso de engenharia elétrica. Passamos por muitos momentos difíceis. Vocês são a parte da Unb que eu mais sentirei falta. Acredito que fiz amizades que duraram até o fim da vida. Esses últimos 6 anos foram incríveis. Obrigado por tudo. Gostaria de agradecer aos amigos que fiz no caratê. Nesses 14 anos de treinos, vocês foram responsáveis por manter minha cabeça no lugar certo.*

*Por último, mas não menos importante gostaria de agradecer a todos os professores que tive. Aprendi muito com os senhores. Dentre esses incríveis professores, gostaria de destacar os meus orientadores. Obrigado prof. Mylène por todos os ensinamentos durante esses últimos três anos. Obrigado por aceitar a minha ideia de projeto mesmo não sendo um projeto da sua área. Foram muitas complicações, mas deu tudo certo. Obrigado por toda atenção que você me deu mesmo morando tão longe. Ser seu orientando foi um enorme prazer. Obrigado prof. Daniel por aceitar ser meu orientador. Obrigado por aceitar esse desafio comigo e pelos muitos ensinamentos sobre aprendizado de máquina.*

*André Henrique Macedo da Costa*

---

## RESUMO

As tatuagens são uma forma de modificação corporal baseada na inserção de tintas na pele que alteram a sua coloração. Os procedimentos de tatuagens têm sido realizados a milhares de anos com os primeiros registros de múmia tatuadas datando do quarto milênio a.C.. Hoje, a tatuagem tem a função de auto identificação. As pessoas buscam tatuagens com o objetivo de expressar sua identidade interior. Logo, há uma relação de unicidade em cada tatuagem. Além disso, a tatuagem recebeu o status de arte, sendo o tatuador considerado um pintor que utiliza a pele humana como meio de expressão. O crescimento das mídias sociais aumentou o compartilhamento de imagens de tatuagens e conseqüentemente o seu plágio, comprometendo o tatuado e o tatuador.

Este trabalho propõe a utilização de rede neurais convolucionais para a identificação da autoria de tatuagens com o objetivo de proteger o artista. O sistema proposto classifica as imagens a partir das características extraídas das imagens de trabalhos passados de sete artistas. No desenvolvimento desse sistema, criou-se um banco de dados contendo os trabalhos de diferentes tatuadores e o teste de cinco diferentes arquiteturas de classificação. O banco de dados contém mais de 1.800 imagens de 7 diferentes tatuadores de 4 diferentes estilos. Fez-se o uso de Redes Neurais Convolucionais, uma técnica de Aprendizado de Máquina, para a extração das características e classificação das imagens. Utilizou-se uma rede já treinada para a extração das características básicas e adicionaram-se camadas a essa rede que foi treinada para realizar a classificação. A rede se mostrou capaz de realizar corretamente a identificação de cada autor. A arquitetura com o melhor desempenho classificou corretamente, em média, 83,7% das imagens dos sete artistas e teve um F1score total de 83. Assim, conclui-se que o procedimento se mostrou eficaz neste contexto.

---

## ABSTRACT

Tattoos are a form of body modification characterised by the insertion of ink into the skin. Tatting procedures have been done for thousands of years with the evidences of tattooed mummies from the fourth millenium BC. Recently, tattoo has received the status of art and tattoo artists are considered painters that use skin as a canvas. But, with the growth of image sharing in social media, the number of cases of tattoo plagiarism has also grown, which affects both the tattoo owner and the tattoo artist. This project proposes a technique to identify the authorship of tattoos aiming to protect the tattoo artist. This system classifies images of seven selected tattoo artists using a set of features extracted from previous works of each artist. During the development of this project, a dataset of tattoo images was created and five different classifying algorithms were tested. The dataset has more than 1,800 images of 7 artist with 4 different styles. Convolutional neural networks were used for the features extraction and classification of the images. A pre-trained network was used as the feature extractor. Extra layers were added to the pre-trained

network for the classification task. The network was successfully identified the artists. The best performing architecture correctly classified on average 83.7% of the total images and had an overall F1score of 83. Thus, we conclude that the procedure proved to be effective in this context.

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	2
1.4	APRESENTAÇÃO DO MANUSCRITO	2
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>3</b>
2.1	APRENDIZADO DE MÁQUINAS	3
2.2	REDES NEURAIS CONVOLUCIONAIS	3
2.3	APRENDIZADO	7
2.4	MÉTRICAS DE DESEMPENHO	9
2.4.1	ACURÁCIA	10
2.4.2	PRECISÃO	10
2.4.3	REVOCAÇÃO	11
2.4.4	F-SCORE	11
<b>3</b>	<b>Solução Proposta</b>	<b>12</b>
3.1	MONTAGEM DO BANCO DE DADOS	12
3.1.1	AQUISIÇÃO DAS IMAGENS	12
3.1.2	HIGIENIZAÇÃO DO BANCO DE DADOS	14
3.1.3	DIMENSIONAMENTO	15
3.2	TREINAMENTO DO CLASSIFICADOR	16
<b>4</b>	<b>Resultados Experimentais</b>	<b>21</b>
4.1	IDENTIFICAÇÃO DE TATUADORES	21
4.1.1	EXPERIMENTO COM CINCO CLASSES	24
4.1.2	EXPERIMENTO COM SEIS CLASSES	28
4.1.3	EXPERIMENTO COM SETE CLASSES	31
4.1.4	DISCUSSÃO DOS RESULTADOS	35
<b>5</b>	<b>Conclusões</b>	<b>37</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>39</b>



# LISTA DE FIGURAS

2.1	Ilustração das semelhanças entre o neurônio e o neurônio artificial. ....	4
2.2	Arquitetura básica de uma rede convolucional artificial .....	5
2.3	Figura com exemplos de filtros utilizados em redes neurais convolucionais.....	6
2.4	Ilustração da operação de Maxpooling em uma imagem. ....	7
2.5	Arquitetura padrão de uma rede neural composta por perceptrons. A arquitetura inclui uma camada de entrada, duas camadas internas e uma camada de saída. ....	8
3.1	Amostras dos diversos artistas presentes no Banco de Dados .....	14
3.2	Amostras dos diversos tipos de padrões indesejados encontrados .....	15
3.3	Arquitetura da VGG16 sem a Camada Final .....	18
3.4	Diagrama do Modelo .....	19
3.5	Exemplo de Arquitetura com Camadas Classificadoras.....	20
4.1	Gráfico do desempenho das diferentes arquiteturas a partir do número de classes. ...	23
4.2	Matriz de Confusão da arquitetura FC1024 para o caso de classificação de cinco classes.....	25
4.3	Matriz de Confusão da arquitetura FC2048 para o caso de classificação de cinco classes.....	26
4.4	Matriz de Confusão da arquitetura FC4096 para o caso de classificação de cinco classes.....	26
4.5	Matriz de Confusão da arquitetura FC8192 para o caso de classificação de cinco classes.....	27
4.6	Matriz de Confusão da arquitetura Linear para o caso de classificação de cinco classes.	27
4.7	Matriz de Confusão da arquitetura FC1024 para o caso de classificação de seis classes.	29
4.8	Matriz de Confusão da arquitetura FC2048 para o caso de classificação de seis classes.	29
4.9	Matriz de Confusão da arquitetura FC4096 para o caso de classificação de seis classes.	30
4.10	Matriz de Confusão da arquitetura FC8192 para o caso de classificação de seis classes.	30
4.11	Matriz de Confusão da arquitetura Linear para o caso de classificação de seis classes.	31
4.12	Matriz de Confusão da arquitetura FC1024 para o caso de classificação de sete classes.	33
4.13	Matriz de Confusão da arquitetura FC2048 para o caso de classificação de sete classes.	33
4.14	Matriz de Confusão da arquitetura FC4096 para o caso de classificação de sete classes.	34
4.15	Matriz de Confusão da arquitetura FC8192 para o caso de classificação de sete classes.	34
4.16	Matriz de Confusão da arquitetura Linear para o caso de classificação de sete classes.	35

# LISTA DE TABELAS

2.1	Tabela contendo as diferentes combinações entre as predições de uma modelo de classificação e o resultado correto dessa mesma predição .....	10
3.1	Lista de artistas por cada estilo do banco de dados .....	13
3.2	Número de Imagens em cada Classe. ....	16
4.1	Tabela contendo os valores da acurácia e o valores desvio padrão dos valores da acurácia para diferentes arquiteturas e diferentes números de classes.....	23
4.2	Tabela contendo as médias das métricas de precisão, revocação e F-score para as arquiteturas propostas no Experimento com Cinco Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação. ....	25
4.3	Tabela contendo as médias das métricas de Precisão, Revocação e F-score para as arquiteturas propostas no Experimento com Seis Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação. ....	28
4.4	Tabela contendo as médias das métricas de Precisão, Revocação e F-score para as arquiteturas propostas no Experimento com Sete Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação. ....	32
4.5	Tabela com os número de parâmetros atualizáveis de cada arquitetura. ....	36
4.6	Tabela contendo o número de parâmetros de diferentes modelo de rede convolucionais utilizadas na classificação de imagens. ....	36

# LISTA DE SÍMBOLOS

## Símbolos Gregos

$\theta$	Parâmetro da rede neural
$\Delta$	Varição de uma grandeza
$\rho$	Taxa de momento da média dos gradientes
$\eta$	Taxa de aprendizado da rede neural
$\epsilon$	Constante do elemento de momento da média dos gradientes

# Capítulo 1

## Introdução

### 1.1 Contextualização

Tatuagem é uma forma de arte e procedimento de mudança corporal no qual ocorre a inserção de tintas na pele. A inserção dos pigmentos altera a cor da pele definitivamente. Os primeiros registros de tatuagens datam do quarto milênio a.C.. Esse registro é de uma múmia encontrada na Itália apresentando padrões feitos com pigmentos na pele. Existem outros registros da presença da tatuagens em diversas culturas em diversas partes do globo em períodos coincidentes. Esses registros estão presentes tanto em populações isoladas como em civilizações do continente americano e do continente asiático. As tatuagens têm sido utilizadas por milhares de anos para diversas funções, incluindo tratamentos medicinais e estética corporal. Um dos usos da tatuagem foi a marcação de escravos e criminosos. Esse uso é, em parte, responsável pela má reputação dessa forma de arte.

Mais particularmente no ocidente, as tatuagens tornaram-se mais populares com as viagens marítimas feitas pelos europeus no século XVIII. Porém, a prática não se fixou devido à dor que o procedimento causava. No fim do século XIX, é inventada a primeira máquina elétrica de tatuagem. Após esse ponto, é visto um crescimento constante do número de adeptos de tatuagens. Atualmente, no século XXI, a tatuagem é uma forma de arte estabilizada com um forte mercado. A tatuagem é muito utilizada como uma forma de expressão de identidade dos indivíduos.

Em suma, apesar das tatuagens serem sido utilizadas há milhares de anos, na última década houve uma explosão da sua popularidade. Essa popularidade também trouxe conflitos quanto aos direitos autorais. Considerando a pele como um "meio de expressão", diversos artistas defendem seus direitos sobre a propriedade autoral das tatuagens por eles criadas. Os processos jurídicos mais conhecidos são entre tatuadores e grandes empresas [1, 2].

## 1.2 Definição do problema

Como já mencionado, junto com a crescente popularidade, vieram diversos problemas relacionados às tatuagens. Um desses problemas é o plágio. O plágio de tatuagens fere tanto o artista como a pessoa portadora da tatuagem. Os tatuadores, assim como outros artistas, dedicam incontáveis horas de trabalho no processo criativo, execução e finalização da tatuagem. Os clientes desses artistas, por sua vez, contribuem com o processo de criação da tatuagem. Muitos daqueles que procuram um tatuador querem uma obra de arte única. Além disso, muitas pessoas fazem tatuagens por razões sentimentais.

Nos últimos anos, houve um grande aumento da quantidade do número de imagens produzidas e compartilhadas. Em grande parte, esse cenário ocorreu pelo surgimento de redes sociais. Essas plataformas digitais contribuíram muito com o plágio de tatuagens. Alguns casos de plágio foram levados à justiça. Esses casos apresentam uma característica em comum, são processos que envolvem artistas e grandes empresas. Os mais famosos envolvem tatuagens feitas em atletas famosos. Curiosamente, não há muitos casos na justiça de plágio de tatuagens envolvendo pessoas físicas. Atribui-se essa falta de processos a uma carência de técnicas que permitam a identificação desses plágios.

## 1.3 Objetivos do projeto

Esse trabalho tem como objetivo apresentar um algoritmo capaz de identificar a autoria de uma tatuagem. O algoritmo deve ser capaz de identificar as diversas características intrínsecas de cada autor. O algoritmo, que é baseado em técnicas de aprendizado de máquinas, utilizando imagens dos próprios artistas para o treinamento e teste.

## 1.4 Apresentação do manuscrito

No capítulo 2, é feita uma revisão bibliográfica sobre o tema de estudo. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento do projeto. Resultados experimentais são discutidos no capítulo 4, seguido das conclusões no capítulo 5. Os anexos contêm material complementar.

## Capítulo 2

# Revisão Bibliográfica

Neste capítulo serão apresentados os conceitos teóricos necessários para o entendimento dessa monografia. Primeiramente, será feita uma breve introdução a área de Aprendizado de Máquina, incluindo conceitos de Rede Neurais Convolucionais. Por fim serão apresentadas as métricas de qualidade que serão utilizadas para medir o desempenho do algoritmo.

### 2.1 Aprendizado de Máquinas

Aprendizado de Máquinas é a área da Computação que permite computadores aprenderem a executar tarefas sem serem programados especificamente para executar aquela tarefa. Para que tal feito seja atingido é necessário que o algoritmo passe por uma fase de treinamento antes de poder executar uma tarefa.

O treinamento pode ser dividido em dois tipos: supervisionado e não supervisionado [3]. Aprendizado supervisionado é aquele onde se apresentam as respostas corretas para o modelo. Aprendizado não supervisionado, ao contrário, ocorre sem a apresentação das respostas corretas ao modelo. Em ambos os cenários, a tarefa do modelo é procurar diferentes conexões entre a base de dados apresentada. Nesse trabalho, iremos focar numa classificação supervisionada.

Em um cenário de classificação, como é o caso desse projeto, a base de dados já é apresentada dividida entre as diferentes classes. Durante o treinamento, há apresentação dos dados e o algoritmo aprende as características de cada uma das classes dessa base de dados.

### 2.2 Redes Neurais Convolucionais

Para que o modelo possa aprender é necessário que se tenha atributos nos quais suas decisões possam se basear. Os atributos são retirados da base de dados previamente catalogadas. Redes Neurais Artificiais são algoritmos de aprendizado de máquina que visam reproduzir o comportamento do cérebro humano. A base para o funcionamento da rede é o comportamento do neurônio [4]. As sinapses em neurônios biológicos ocorrem quando um estímulo elétrico é capaz de vencer a

barreira de potencial do neurônio. O neurônio artificial visa modelar esse comportamento através de uma função não linear. O neurônio biológico não permite a passagem do estímulo elétrico caso esse possua potência inferior ao valor da sua barreira de potencial do neurônio. O Neurônio Artificial não permite a passagem da informação caso ela não exceda um determinado valor. O neurônio orgânico salva o valor do limiar de excitação na sua composição química, alterando assim a sua impedância. Os Neurônios Artificiais armazenam seu limiar de excitação na forma de pesos na função não linear. Essa função não linear recebe o nome de função de ativação. As funções de ativação possuem uma não linearidade que divide dois patamares na função. Essa diferença entre os patamares funcionará com a barreira de potencial do neurônio. Ela servirá para discriminar a não linearidade existente entre a informação, que gostaríamos de classificar, e a informação que deve ser desprezada. A Figura 2.1 ilustra a semelhança entre os comportamentos do neurônio e do neurônio artificial.

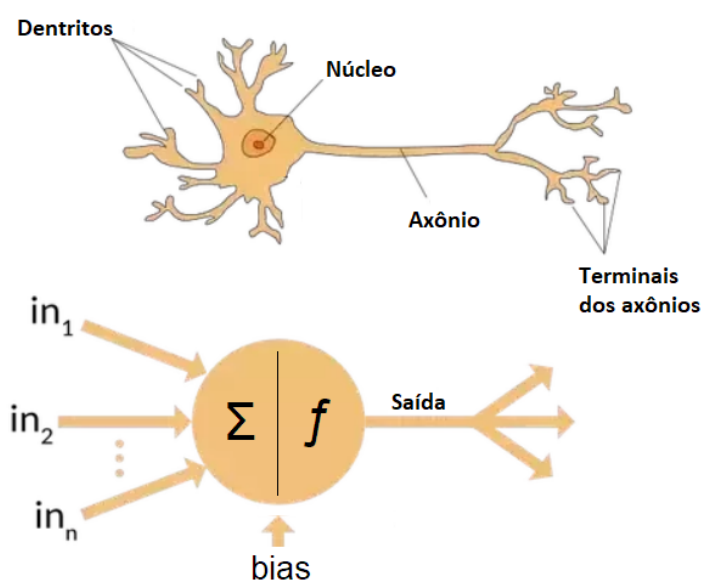


Figura 2.1: Ilustração das semelhanças entre o neurônio e o neurônio artificial.

Rede neurais convolucionais, do inglês *Convolutional Neural Networks* (CNN), são um tipo específico de rede neural artificial. Elas possuem a adição de um tipo especial de camada, as camadas convolucionais. De forma geral, as CNNs possuem uma estrutura básica formada por camadas convolucionais e camadas de *pooling*. As camadas convolucionais extraem os dados das imagens. As camadas de *pooling* fazem a sub-amostragem da informação. Por último, os dados passam por camadas densas que executam a classificação utilizando as características extraídas pelas camadas supracitadas. As camadas densas são conjuntos de unidades menores chamadas perceptrons. A capacidade de processamento da rede é resultado do empilhamento dessas diferentes camadas [5, 6]. A quantidade de camadas é define a profundidade da rede. Rede profundas são capazes de realizar a identificação de padrões com alto detalhamento, como é o caso da identificação de obras de arte [7, 8, 9, 10]. A Figura 2.2 apresenta a arquitetura básica de uma rede neural artificial convolucional.

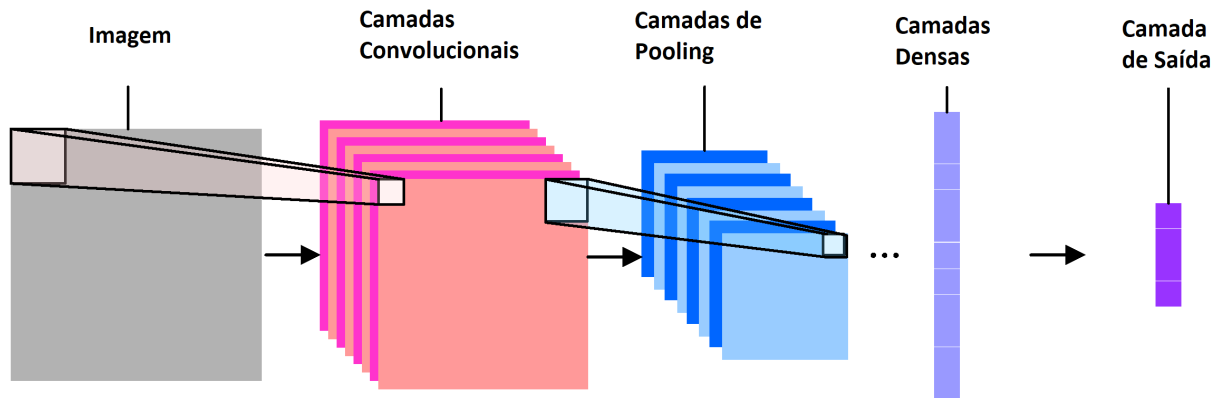


Figura 2.2: Arquitetura básica de uma rede convolucional artificial

As camadas convolucionais realizam convoluções entre a imagem e os diversos filtros da rede. A operação de convolução é amplamente utilizada no processamento de imagens [11, 12, 13]. As camadas convolucionais utilizam um conjunto de filtros para extrair características das imagens. Esses filtros discriminam a orientação e a textura de imagens. Seu comportamento é inspirado no córtex visual dos seres humanos, que é a camada mais externa do cérebro e é responsável pelo processamento neural mais sofisticado. O córtex visual é a parte do córtex que recebe os estímulos oriundos dos olhos e os processa, sendo responsável pelo processamento neural mais sofisticado. Quando utilizados em operações de convolução, os filtros discriminam a orientação dos elementos visuais presentes na imagem. Por um exemplo, um filtro com orientação horizontal evidenciará as diversas estruturas horizontais presentes numa imagem, enquanto um filtro com orientação vertical evidenciará as informações verticais da imagem.

A operação de convolução 2-D é definida pela seguinte equação:

$$Y[x, y] = F[x, y] \otimes G[x, y] = \sum_{n_1=0}^N \sum_{n_2=0}^N F(n_1, n_2)G(x - n_1, y - n_2) \quad (2.1)$$

na qual  $x$  e  $y$  correspondem aos índices horizontais e verticais da imagem  $F[x, y]$  e do filtro  $G[x, y]$  e  $n_1$  e  $n_2$  são, respectivamente, os índices de deslocamento horizontal e vertical da operação de convolução.

As diferentes orientações projetadas pelos filtros permitem que as camadas convolucionais detectem características independentemente da sua orientação espacial na imagem. Na operação de convolução, o filtro realiza uma varredura na imagem. Podendo assim encontrar a mesma característica visual múltiplas vezes na mesma imagem, o que pode contribuir para a eficiência no que diz respeito ao número de parâmetros. Por último, vale ressaltar que os filtros presentes nas camadas convolucionais levam em consideração os pixels vizinhos. Pixels vizinhos carregam informação com maior correlação do que pixels distantes [11]. Dessa forma, o produto final da convolução carrega mais informação. Como já dito, há uma relação entre o número de camadas e a capacidade de representação da CNN. Redes mais profundas são capazes de realizar a identificação de padrões com alto detalhamento, como é o caso da identificação de obras de arte [7, 8, 9, 10]. A



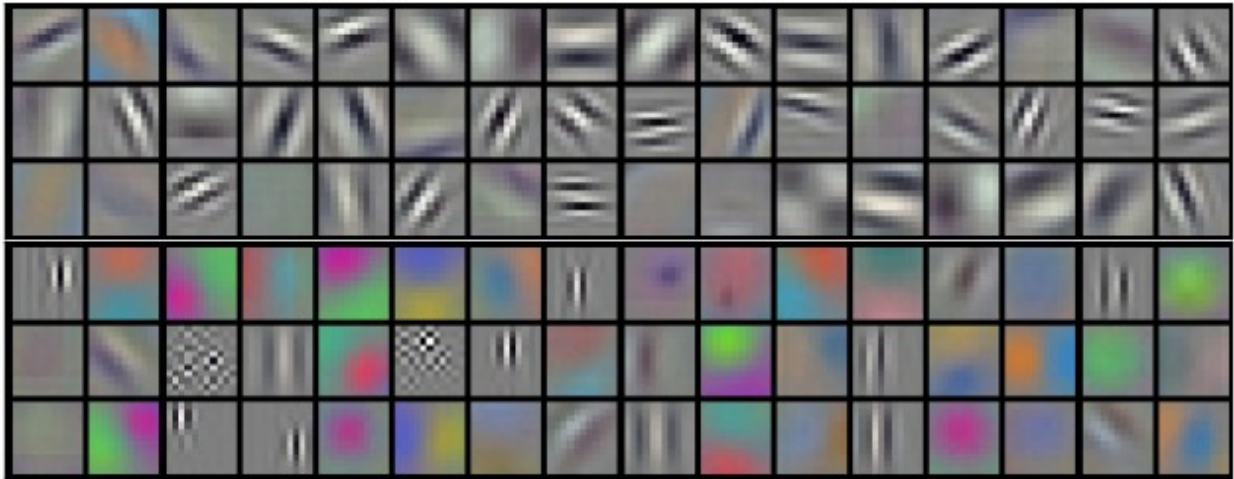


Figura 2.3: Figura com exemplos de filtros utilizados em redes neurais convolucionais.

profundidade da rede impõe mudanças nos filtros. Os filtros mais internos da rede se tornam mais complexos, extraíndo características mais definidas e visualmente menos abstratas. Dessa forma, cria-se uma hierarquização das características extraídas. Os filtros presentes no fim da rede irão se assemelhar mais com o conteúdo da imagem, permitindo que a rede se adapte à identificação de padrões visuais mais complexos.

Vale salientar que nem sempre é necessário aplicar a convolução sobre todos os pixels de uma imagem. A varredura do filtro pode ser espaçada. Dessa forma, se diminui a quantidade de informação gerada pelas camadas. O espaçamento é controlado pelo parâmetro passo, também chamado de *Stride*. O passo é o número de pixels que serão pulados durante a execução de cada aplicação do filtro. Se o passo tiver o seu tamanho igual ao tamanho de uma das dimensões do filtro, não haverá interseção entre as regiões onde o filtro é aplicado.

As camadas convolucionais geram uma quantidade muito grande de dados, mas nem todos os dados gerados possuem informação relevante para a execução da tarefa. Dependendo da imagem que é apresentada para o modelo, vários filtros não produzem mapas com muita informação. Para se aumentar a eficiência do modelo, é necessário filtrar os mapas gerados pelas camadas convolucionais. As camadas de *pooling* desempenham essa função. Elas diminuem o tamanho do mapa de características produzidos nas camadas de convolução. Fazem assim uma forma de subamostragem, que é muito importante para o desempenho do algoritmo porque elimina parte da computação desnecessária.

O *pooling* ocorre através de uma máscara que é passada pelo mapa de características. É possível controlar as dimensões da máscara e o tamanho do passo efetuado pela máscara. As dimensões das máscaras em geral são de 2x2 ou 3x3 pixels. O tamanho do passo geralmente acompanha as dimensões de cada máscara. Sendo assim não se costuma fazer a sobreposição das regiões filtradas. Existem diferentes métodos de *pooling*. O *Maxpooling* executa a função de máximo local entre os valores dos pixels dentro da máscara. Em um caso onde se utiliza uma máscara 2x2 com passo 2, a imagem resultante teria a metade do tamanho original e cada pixel gerado terá o valor do pixel com maior valor daquela dada região. Além de compactar os dados, a camada de *pooling* tem a

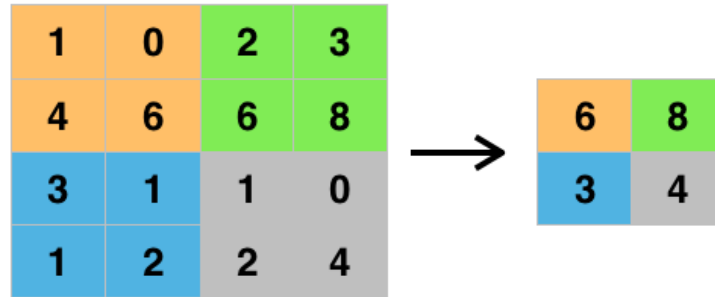


Figura 2.4: Ilustração da operação de Maxpooling em uma imagem.

capacidade de generalizar a informação contida no mapa de características. A Figura 2.4 ilustra uma simplificação da operação de *maxpooling* em uma imagem.

As camadas densas também podem ser chamadas de perceptron multicamada. Elas são compostas de neurônios artificiais, Perceptrons. Eles são a forma mais básica de construção de camadas nas Redes Neurais. São amplamente utilizados na classificação binária [14]. Estas camadas têm como entrada um vetor,  $X$ , que é multiplicado por um outro vetor,  $W$ , contendo os pesos daquele perceptron. Além disso, soma-se um termo de bias,  $b$ . A equação do perceptron é definida por:

$$Y = (X \cdot W + b). \quad (2.2)$$

A soma desses termos é utilizada como entrada na função de ativação do perceptron. A função de ativação fará a separação entre a ativação ou a atenuação daquela informação na rede neural. Uma das funções de ativação mais populares é a sigmóide descrita pela seguinte equação:

$$A = \frac{1}{1 + e^{-Y}}. \quad (2.3)$$

O perceptron multicamadas é um agrupamento de perceptrons unitários. Pode-se agrupá-los em paralelo formando uma camada e, por sua vez, podem ser agrupadas em série formando uma rede. A arquitetura das camadas é diretamente responsável pelo tipo de tarefa que o algoritmo consegue executar. Um maior número de camadas pode gerar modelos mais complexos, que são capazes de modelar não-linearidades [4]. A Figura 2.5 apresenta a estrutura de uma rede neural formada por perceptrons organizados em camadas.

## 2.3 Aprendizado

O aprendizado de uma rede neural ocorre por meio de dois processos. O primeiro deles é a propagação direta. Nele, os dados são colocados na entrada do modelo e fazem o caminho direto até a saída do modelo. A propagação direta é, em resumo, a rede em operação. A propagação

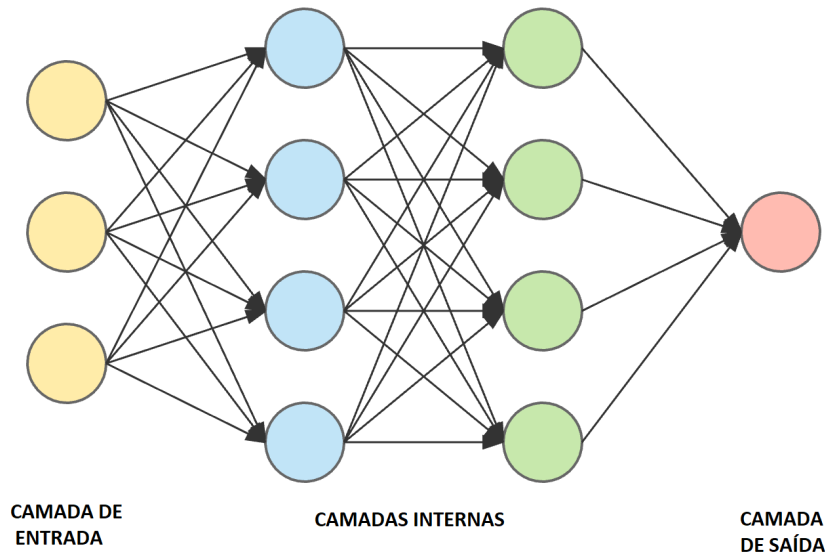


Figura 2.5: Arquitetura padrão de uma rede neural composta por perceptrons. A arquitetura inclui uma camada de entrada, duas camadas internas e uma camada de saída.

reversa é processo de passagem da informação no sentido reverso da rede [14]. Nesse processo, haverá a atualização dos parâmetros da rede. O algoritmo de propagação reversa é de extrema importância para o processo de aprendizagem da rede neural [12]. Pode se dizer que é durante a propagação reversa que a rede está de fato aprendendo a executar uma tarefa. O objetivo da propagação reversa é alterar os valores dos pesos de cada camada. Dessa forma, pode-se mapear corretamente as entradas às saídas. Existem diferentes algoritmos que podem ser utilizados para realizar as etapas da propagação reversa. Neste trabalho são utilizadas a entropia cruzada como função de erro e a RMSProp, do inglês *root mean squared propagation*, como função de otimização [15]. Ambas essas funções serão explicadas a seguir.

No processo de aprendizado, o resultado da propagação direta é comparado ao resultado desejado. Caso, eles sejam iguais não há a necessidade de se alterar os parâmetros da rede. No entanto, se o resultado emitido pelo modelo for diferente do resultado correto, deve-se fazer a atualização dos parâmetros das camadas da rede. Os resultados são comparados através de uma função de erro. Essa função retorna uma medida entre a diferença do resultado esperado e o resultado obtido. A função de erro é dada por:

$$E = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^m [y_{ij} \log(p_{ij})], \quad (2.4)$$

onde  $i$  é o índice do número de exemplos utilizados,  $y_i$  é a  $i$ -ésima classe de cada exemplo e  $p_i$  é o resultado predito pelo modelo para o  $i$ -ésimo exemplo. Considerando que cada parâmetro contribui em parte com o erro de uma predição, é possível calcular a contribuição desse parâmetro para o erro total. A saída de cada camada da rede neural é a entrada da próxima camada, ou seja, pode-se utilizar a regra da cadeia para se encontrar a derivada parcial do erro com relação a cada parâmetro da rede. A equação da contribuição de cada erro é dada por:

$$J(\theta_{t,k}) = \frac{\partial E}{\partial \theta_{t,k}}, \quad (2.5)$$

sendo  $k$  o índice dos diferentes parâmetros  $\theta$  da rede em cada instante  $t$  do treinamento.

Utiliza-se a contribuição de cada parâmetro e calcula-se o gradiente com relação ao parâmetro passado. Esse gradiente irá apontar a direção para a qual os valores dos parâmetros devem se alterar para que se diminua o erro de classificação. O gradiente pode ser escrito como sendo:

$$g_{t,k} = \nabla_{\theta} J(\theta_{t,k}), \quad (2.6)$$

onde o gradiente  $g_{t,k}$  é o gradiente de cada parâmetro  $\theta$  com índice  $k$  em um instante  $t$  do processo de treinamento.

O método RMSprop não aplica o gradiente diretamente ao parâmetro. Esse método realiza uma suavização do ritmo de aprendizado de forma a diminuir o tempo total do aprendizado. A suavização é realizada por meio da média dos quadrados dos gradientes passados. Essa média é descrita por:

$$M[g^2]_t = \rho M[g^2]_{t-1} + (1 - \rho)g_t^2, \quad (2.7)$$

onde  $M[g^2]_{t-1}$  é a média dos quadrados dos gradientes passados,  $(g_{t-1})$  e  $\rho$  é uma constante que reduz a variação da média dos gradientes. A constante  $\rho$  pode ser definida como uma inércia do sistema. A variação dos parâmetros é empregada considerando o último gradiente multiplicado pela taxa de aprendizado e reduzido pela média dos gradientes passados. A variação que será aplicada à cada parâmetro,  $\theta$  é dada por:

$$\Delta\theta_{t,k} = -\frac{\eta}{\sqrt{M[g^2]_t + \epsilon}}g_t, \quad (2.8)$$

onde  $\eta$  é a taxa de aprendizado e  $\epsilon$  é uma constante de valor muito pequeno que impede uma divisão por zero. O valor do novo parâmetro  $\theta_{t+1}$  é dado por:

$$\theta_{t+1} = \theta_t + \Delta\theta_t = \theta_t - \frac{\eta}{\sqrt{M[g^2]_t + \epsilon}}g_t. \quad (2.9)$$

## 2.4 Métricas de Desempenho

Nesta seção, serão apresentadas as métricas utilizadas durante a avaliação de desempenho do modelo. As métricas utilizadas são:

- Acurácia
- Precisão
- Revocação
- F-score

É importante que se utilize mais de uma métrica, uma vez que métricas diferentes se complementam e têm a capacidade de mostrar o real desempenho do modelo estudado. O uso de apenas uma métrica pode esconder algum desempenho não satisfatório.

Em aplicações de classificação, os resultados podem ser divididos em 4 diferentes classes. Essas classes são a combinação entre o resultado esperado e o resultado predito pelo modelo. Em aplicações de classificação de uma determinada classe, as predições podem ser classificadas como positivas, caso ela pertença aquela classe, ou negativa, caso o contrário. Os verdadeiros positivos, VP, são os casos onde o modelo prediz o exemplo como positivo e ele, de fato, é positivo. O falso positivo, FP, seria o caso contrário, onde o modelo prediz um exemplo falso como positivo. A mesma análise é feita para as predições negativas. Falsos negativos, FN, são os casos positivos preditos como negativos e os verdadeiros negativos, VN, são os exemplos negativos preditos como negativos. A análise das métricas de um algoritmo de classificação pode ser medido através dessas 4 classes. A Tabela 2.1 ilustra os conceitos explicados acima.

		Predito pelo modelo	
		Positivo	Negativo
Classes Reais	Positivo	VP	FN
	Negativo	FP	VN

Tabela 2.1: Tabela contendo as diferentes combinações entre as predições de uma modelo de classificação e o resultado correto dessa mesma predição

### 2.4.1 Acurácia

A acurácia é a métrica que retorna o número de acertos levando em consideração todas as tentativas. Em um cenário de classificação de imagens, ela seria o número de imagens classificadas corretamente dividida pelo número total de imagens, TE.

$$Ac = \frac{VP + VN}{TE}, \quad (2.10)$$

A acurácia é uma métrica que revela o percentual de acertos do modelo. Geralmente, é a primeira métrica a ser utilizada em algoritmos de aprendizado de máquinas. Porém, não é capaz de descrever corretamente o desempenho do modelo. Em aplicações com um número desequilibrado de exemplos em cada classe, a acurácia perde sua relevância tendo em vista que a classificação de apenas uma das classes renderia uma acurácia semelhante ou até superior. Esse fenômeno é conhecido como Paradoxo da Acurácia [16].

### 2.4.2 Precisão

A Precisão é definida como:

$$Pr = \frac{VP}{VP + FP}. \quad (2.11)$$

Esta métrica revela a capacidade do algoritmo de identificar os verdadeiros positivos entre os preditos como positivos. Em outras palavras, a precisão fornece a porcentagem de acerto que realmente são acertos. Essa métrica é capaz de evidenciar a presença de falsos positivos.

Um modelo com uma alta precisão, é capaz de identificar os diferentes detalhes de uma classe. Uma precisão alta indica que o modelo não classificou muitos exemplos como falsos positivos. uma precisão baixa indica que o modelo classificou um número grande de exemplos como falsos positivos. Nesse cenário, o modelo é incapaz de descrever detalhadamente a classe. Por sua vez, identificando uma classe mais abrangente que engloba a classe desejada e classes indesejadas.

A precisão é importante em aplicações onde não se pode ter falsos positivos. Aplicações de aprendizado de máquinas como exame médico necessitam de alta precisão. Em caso de diagnóstico de doença graves, como câncer, em pacientes sadios pode levar a execução de tratamentos desnecessários. No caso de identificação de obras de arte, o efeito de falsos positivos não é tão severo.

### 2.4.3 Revocação

A revocação mede a capacidade do modelo de encontrar verdadeiros positivos, sendo calculada pela seguinte equação:

$$Re = \frac{VP}{VP + FN}. \quad (2.12)$$

Essa métrica calcula a porcentagem dos exemplos verdadeiros que realmente foram classificados como verdadeiros e o número de exemplos verdadeiros que o modelo não pode identificar. Em um algoritmo de classificação uma alta revocação indica que o modelo percebeu a maioria dos exemplos positivos como positivos, o que indica uma alta capacidade do modelo de descrever corretamente grande parte das características daquela classe. O caso contrário, quando o modelo possui baixa revocação, indica que o modelo teve muitos exemplos falsos positivos. Neste cenário, o modelo não estaria identificando um número desejado de características da classe em questão.

### 2.4.4 F-score

A métrica *F-score* é a média harmônica entre a precisão e a revocação do do modelo, que é dada pela seguinte equação:

$$\begin{aligned} F - score &= \frac{Re^{-1} + Pr^{-1}}{2} = \\ &= 2 \frac{Pr \cdot Re}{Pr + Re} = \\ &= \frac{2(VP)}{2(VP) + FP + FN}. \end{aligned} \quad (2.13)$$

A métrica em si não possui significado do desempenho do modelo já que é a média entre duas outras métricas. Porém, essa métrica retorna uma visão mais ampla do desempenho do modelo [17, 18]. A *F-score* possui valor entre 0 e 1. Para que o valor máximo (1) possa ser atingido, não deve haver ocorrências de falsos positivos e falsos negativos. Um modelo que consiga alcançar esse medida seria um classificador perfeito, classificando apenas verdadeiros positivos e verdadeiros negativos.

## Capítulo 3

# Solução Proposta

Esse capítulo tratará da metodologia implementada nesse projeto. Serão aprofundadas as etapas utilizadas para a criação do Identificador de Autoria de Tatuagens. O processo está dividido nas etapas abaixo:

- Montagem do Banco de Dados
- Treinamento do Classificador

### 3.1 Montagem do Banco de Dados

#### 3.1.1 Aquisição das Imagens

Primeiramente, foi realizada uma pesquisa sobre a existência e a disponibilidade de banco de dados sobre tatuagens. Encontrou-se o banco de dados Tatt-C [19]. Tatt-C é um banco de dados pertencente ao desafio de mesmo nome promovido pelo NIST, *National Institute of Standards and Technology*. Apesar de grande, o banco de dados não possui as informações dos autores das tatuagens. O NIST disponibiliza a base de dados com o objetivo de facilitar a criação de algoritmos de reconhecimento de tatuagens, tendo como foco as aplicações forenses.

Uma vez que não foi possível encontrar uma base de dados já existente, foi necessário criar uma. Infelizmente, não conseguimos apoio de tatuadores locais. Muitos não se mostraram receptivos à ideia de compartilhar seu material. Nesse cenário, foi necessário utilizar a internet para a obtenção das imagens. Utilizou-se a rede social Instagram em conjunto com o programa InstaG Downloader. Dessa forma, foi possível entrar no perfil de um artista e baixar as suas imagens públicas.

De posse de um conjunto de imagens de tatuagens, foi possível escolher estilos de tatuagens com uma larga faixa de elementos visuais. Esses diferentes elementos visuais têm como objetivo desafiar a arquitetura da rede convolucional. Foram escolhidos estilos preto-e-branco e coloridos além de estilos com linhas finas e linhas grossas. Segue abaixo a lista com os estilos escolhidos.

- Fine Line

- Water Color
- New Tradicional
- Irezumi

Dos estilos escolhidos, Irezumi é o mais antigo e tradicional dos estilos, possuindo suas raízes no período feudal do Japão. Ele se destaca pela sua estética com animais e criaturas da cultura japonesa. Possui linhas e cores bem fortes e de maneira geral são tatuagens que cobrem grande parte do corpo. Os outros estilos escolhidos são estilos mais recentes. O estilo *New Tradicional* possui elementos gráficos, como linhas e cores, semelhantes ao estilo Irezumi. No entanto, possui conteúdos muito diferentes. Os estilos *Water Color* e *Fine Line* possuem elementos visuais diferentes dos estilos supracitados. Enquanto os estilos Irezumi e *New Tradicional* possuem bordas bem definidas, *Water Color* apresenta transições mais suaves e o *Fine Line* é monocromático. A principal característica do estilo *Fine Line* desse estilo é o pontilhismo empregado no preenchimento das formas. Devido a presença de apenas uma cor de tinta, o controle dos níveis de cinza da tatuagem é feitos através do controle do número de ponto por área. Esse elemento foi escolhido para ser comparado às lentas transições do *Water Color*.

Após a escolha dos estilos se fez a escolha de tatuadores. A escolha dos tatuadores foi feita visando ter mais de um tatuador e ou estilo. Dessa forma, haveria uma maior dificuldade na tarefa de classificação, tornando o algoritmo mais robusto. Essa escolha também permitiu avaliar a arquitetura escolhida para o reconhecimento e classificação do estilo de cada artista. É válido ressaltar que os artistas escolhidos não se limitam a um só estilo, havendo trabalhos em estilos totalmente diferentes do seu estilo principal. Adotou-se os estilos principais de cada artistas como sendo o seu estilo.

Neste trabalho, os artistas Matt Beckerich e Mike Rudenball representam o estilo Irezumi [20, 21], os artistas Krish Trece e Manu Cruz representam o estilo New Tradicional [22, 23]. Escolheu-se os artistas Manu Cruz e Krish Trece devido ao fato de ambos trabalharem no mesmo estúdio. O objetivo era testar o algoritmo com dois artistas com muitas semelhanças. Os artistas Pablo Ortiz e Dynoz representam o estilo *Water Color* [24, 25]. Apesar de representarem um mesmo estilo, a sua interpretações do estilo *Water Color* são distintas. Dynoz utiliza cores mais saturadas e que se assemelham mais aos representantes dos estilos Irezumi e *New Tradicional*. Por fim, a artista Gabriela Arzabe representa o estilo *Fine Line* [26]. A Figura 3.1 apresenta alguns exemplos de imagens de cada tatuador.

Fine Line	Water Color	New Tradicional	Irezumi
Gabriela Arzabe	Pablo Ortiz	Krish Trece	Matt Beckerich
	Dynoz	Manu Cruz	Mike Rudenball

Tabela 3.1: Lista de artistas por cada estilo do banco de dados



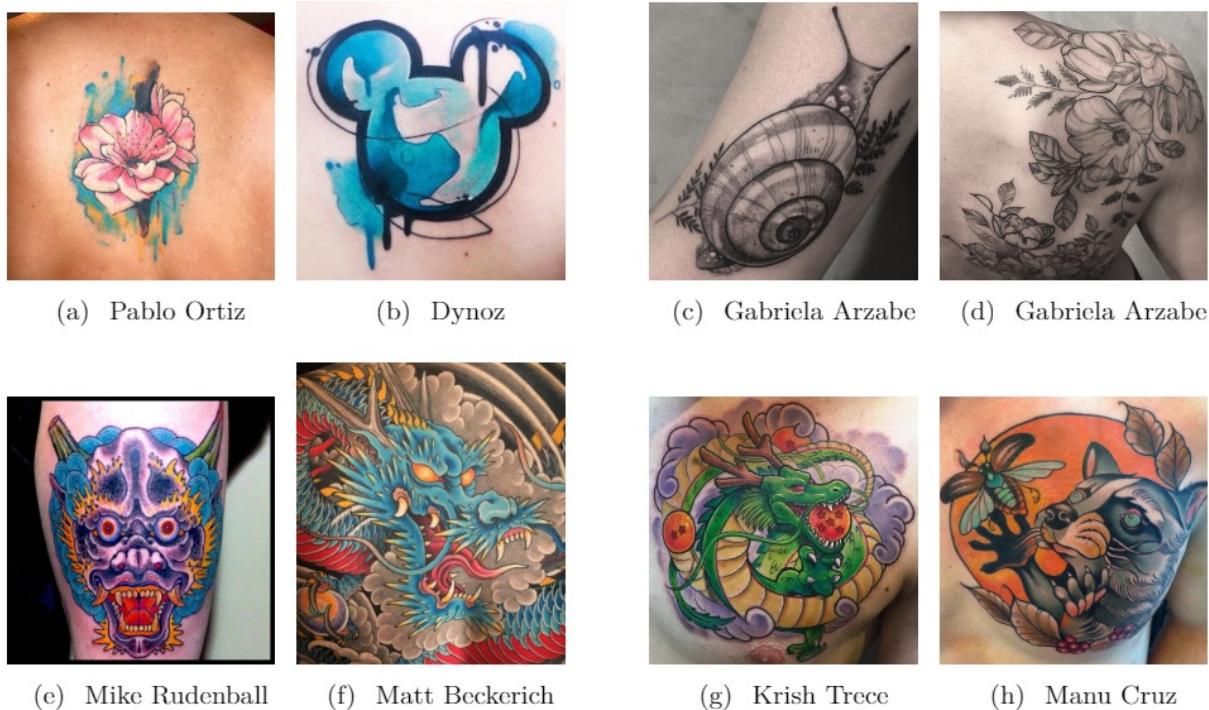


Figura 3.1: Amostras dos diversos artistas presentes no Banco de Dados

### 3.1.2 Higienização do Banco de Dados

Devido, principalmente, à origem das imagens fez-se necessário a higienização do banco de dados. Muitas das imagens obtidas não são imagens de tatuagens. Nas imagens contendo tatuagens, existem partes que não contém a tatuagem e não poderiam ser apresentadas à rede neural. Essas partes indesejadas incluem fundos que apresentam informações que a rede neural poderia utilizar para identificar o artista [7]. Dentre os fundos de imagens mais comuns estão: papéis de parede do estúdio, paredes texturizadas no estúdio, códigos de identificação e textura de objetos, como ilustrado na Figura 3.2. Outro problema é o local da tatuagem. Dependendo da parte do corpo onde a tatuagem se encontra, pode ser difícil remover padrões indesejados. Tatuagens em membros, por exemplo, são mais difíceis de isolar do fundo devido ao seu formato mais longo.

Em suma, foi remover esses padrões visuais para que a rede neural não fizesse a classificação utilizando estas características das fotografias. Esta higienização foi feita manualmente. As etapas adotadas estão enumeradas a seguir:

1. O primeiro passo da higienização foi a eliminação de todas as imagens que não fossem tatuagens. Como as imagens são provenientes de mídias sociais, entre as imagens coletadas havia outros tipos de fotografia. Esse primeiro filtro visou eliminar fotos de pessoas, retratos dos artistas e fotografias de paisagens.
2. O segundo passo foi a eliminação de fotografias com múltiplas tatuagens ou múltiplas vistas da mesma tatuagem. Em caso de um mosaico de várias tatuagens, optou-se por não recortar e produzir diversas imagens, uma vez que muitas das imagens utilizadas na composição



Figura 3.2: Amostras dos diversos tipos de padrões indesejados encontrados

desses mosaicos já estavam presentes no banco de dados. Nos casos onde haviam múltiplas vistas da mesma tatuagem, fez-se a opção de recortar e manter apenas a vista central. As imagens contendo múltiplas vistas geralmente eram de membros superiores e inferiores do corpo humano. A vista central das fotos apresentava o maior número de detalhes.

3. O terceiro passo visou separar ilustrações em papel ou tela e rascunhos de tatuagens. Em geral, as ilustrações em papel ou telas apresentavam grande semelhança com as tatuagens finalizadas, mas sem apresentar trechos com pele. Como já mencionado, as redes neurais convolucionais são capazes de detectar pequenas variações na intensidade luminosa e na variação cromática e reconhecer a textura do material utilizado nas telas de quadros [7]. Como nesse projeto o material de composição da obra de arte é a pele humana as ilustrações em papel e tela foram eliminadas por não apresentarem as mesmas características de uma tatuagem na pele.
4. O quarto passo da higienização foi o corte das tatuagens. O objetivo dessa etapa é recortar as imagens sem retirar partes das tatuagens. As partes retiradas das imagens continham texturas do estúdio como papéis de parede e materiais utilizados no processo da tatuagem. Outras partes indesejadas das imagens eram outras tatuagens. Em alguns casos, foi necessário ponderar se seria melhor cortar partes da tatuagem para eliminar parte das imagens. Nesses casos, foi feita o corte da imagem, retirando parte da tatuagem.

### 3.1.3 Dimensionamento

Após terminar a higienização da base de dados, fez-se a separação das imagens do banco de dados em três diferentes partes: treinamento, validação e teste. Seguiu-se uma prática comum nos trabalhos de aprendizados de máquinas [27]. Alocou-se 70% da base de dados de cada classe para o treinamento, 20% para a validação e 10% para teste.

Sabe-se que o número de exemplos por classe pode impactar o desempenho da rede neural

[28]. De forma semelhante aos neurônios biológicos, as redes neurais artificiais apresentam melhor desempenho na classificação de classes com um maior número de exemplos. A maior quantidade de exemplos permite que o modelo retire mais características daquela classe e melhor ajuste os seus pesos. Por consequência, a rede neural terá menos contato com a classe mais rara, extrairá menos características e terá seus parâmetros mal alocados para a classificação. Utilizou-se a menor classe como base para o número de imagens por classes. Ou seja, os tatuadores com mais imagens tiveram apenas parte das suas imagens utilizadas. Por uma questão de temporalidade, foram consideradas as imagens postadas em um mesmo período. Com isso, diminui-se a chance de haver variações do estilo do autor. A Tabela 3.2 apresenta uma descrição das classes (artistas) e do número de imagens utilizados no treinamento, validação e teste do método.

Classe	Imagens		
	Treinamento	Validação	Teste
Arzabe	168	48	25
KrishTattoo	200	45	24
Matt Beckerich	230	51	20
Pablo Ortiz	200	56	28
Mike Ridenball	195	55	27
Manu Cruz	138	40	20
Dynoz	202	56	25

Tabela 3.2: Número de Imagens em cada Classe.

É uma prática comum nos trabalhos de classificação de imagens com rede neurais convolucionais se dividir a base de dados aleatoriamente em cada sessão de treinamento do algoritmo. Essa medida tem por objetivo aumentar a aleatoriedade do treinamento, uma vez que um treinamento mais diverso teria a capacidade de inferir maior capacidade de generalização. Nesse trabalho, não será feito o embaralhamento da base de dados. As consequências dessa escolha serão discutidas no Capítulo 5. Porém já pode ser dito que se perde parte do poder de generalização do problema.

## 3.2 Treinamento do Classificador

O treinamento do modelo de classificação ocorreu em linguagem Python. A biblioteca utilizada para a execução do treinamento foi a Keras. A biblioteca já possui ferramentas de pré-processamento das imagens, além de modelos treinados na base de dados Imagenet. Não foi utilizada nenhuma forma de pré-processamento além da higienização feita durante a formação do banco de dados. Houve o redimensionamento das imagens para que pudessem ser apresentadas à rede. O redimensionamento foi feito para que todas as imagens tivessem tamanho 224x244 pixels.

Utilizou-se o modelo da rede neural VGG16 [29]. Ela já havia sido treinada na base de dados para a classificação de imagens, Imagenet [30]. Em outros trabalhos, é utilizada a técnica de transferência de aprendizado [31], que permite utilizar uma rede pré treinada em uma base

de dados mais abrangente. Nesta técnica, o aprendizado em parte das camadas, é bloqueado e o aprendizado é realizado apenas nas camadas mais profundas da rede. A aplicação de uma rede treinada em bases de dados diferentes é possível graças à transferibilidade das características entre diferentes redes com aplicações correlatas [32]. No nosso trabalho, esse método foi adaptado devido a escassez de dados. O número de imagens por classe de tatuador não é suficiente para a treinar as camadas da VGG16, sendo necessário empregar a VGG16 com as camadas totalmente congeladas. Contudo não foram utilizadas todas as camadas da VGG16. Retirou-se a última camada, que é responsável pela classificação final do modelo. As camadas restantes tiveram o seu treinamento bloqueado, ou seja, não houve a atualização dos pesos das funções de ativação durante o treinamento. O objetivo do bloqueio de treinamento é de utilizar a rede VGG16 como um extrator de características. O extrator irá entregar um vetor com as características extraídas.

As camadas subsequentes são responsáveis pela classificação das imagens. As camadas de classificação recebem as características extraídas pela VGG16. Essas características entregues na forma de um vetor de 100 elementos. Foram utilizadas diferentes configurações de redes classificadoras. A primeira configuração é um classificador linear. O classificador linear possui uma camada, a camada de saída. Essa configuração é a configuração mais próxima da VGG16 original. A única alteração é a troca da camada de saída de 1.000 elementos da ImageNet pela camada com o número de diferentes tatuadores.

As outras configurações testadas possuem mais uma camada oculta. A inclusão dessa camada oculta teve como objetivo aumentar a capacidade da rede de identificar não linearidades. As estruturas com mais de uma camada densa são perceptrons multicamadas. São utilizados diferentes tamanhos de camadas ocultas. Os diferentes tamanhos de camadas são necessários para averiguar se o algoritmo se beneficiaria com a adição de mais uma camada compressiva ou de uma camada expansiva. A Figura 3.3 apresenta a parte utilizada da arquitetura da VGG16, enquanto que a Figura 3.4 apresenta o fluxograma resumido do modelo.

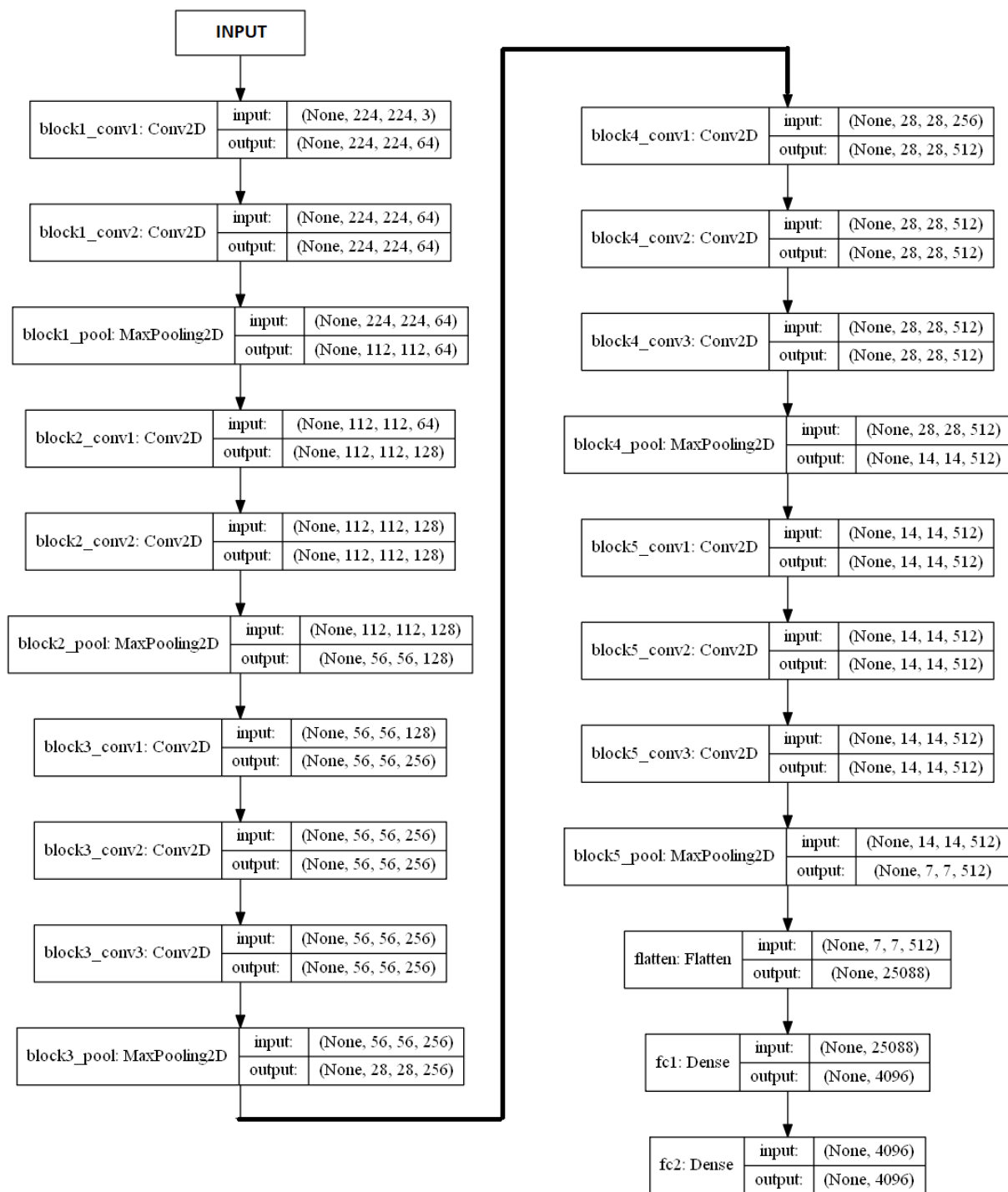


Figura 3.3: Arquitetura da VGG16 sem a Camada Final

As camadas foram escolhidas levando em relação a última camada da VGG16 modificada. Ela possui 4.096 neurônios. Escolheu-se mais camadas compressivas em razão do excessivo número de parâmetros já existente na configuração com a camada de 8192 neurônios. Dessa forma, as chances de sobreajuste, *Overfitting* são reduzidas [33] [34].

Nas Seções seguintes, o modelo onde se substitui apenas a última camada da VGG16 será

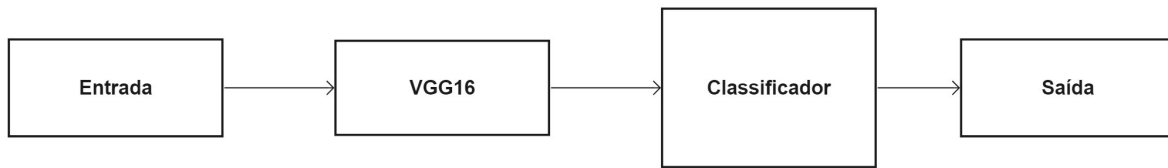


Figura 3.4: Diagrama do Modelo

tratado pelo nome de Linear. As arquiteturas com camadas ocultas serão chamadas pelo código formado pelas letras, FC, e o respectivo número de neurônios da camadas ocultas. Por exemplo, a arquitetura com camada com 2.048 neurônios na camada oculta será tratada pelo nome de FC2048. A Figura 3.5 apresenta um exemplo de arquitetura FC4096.

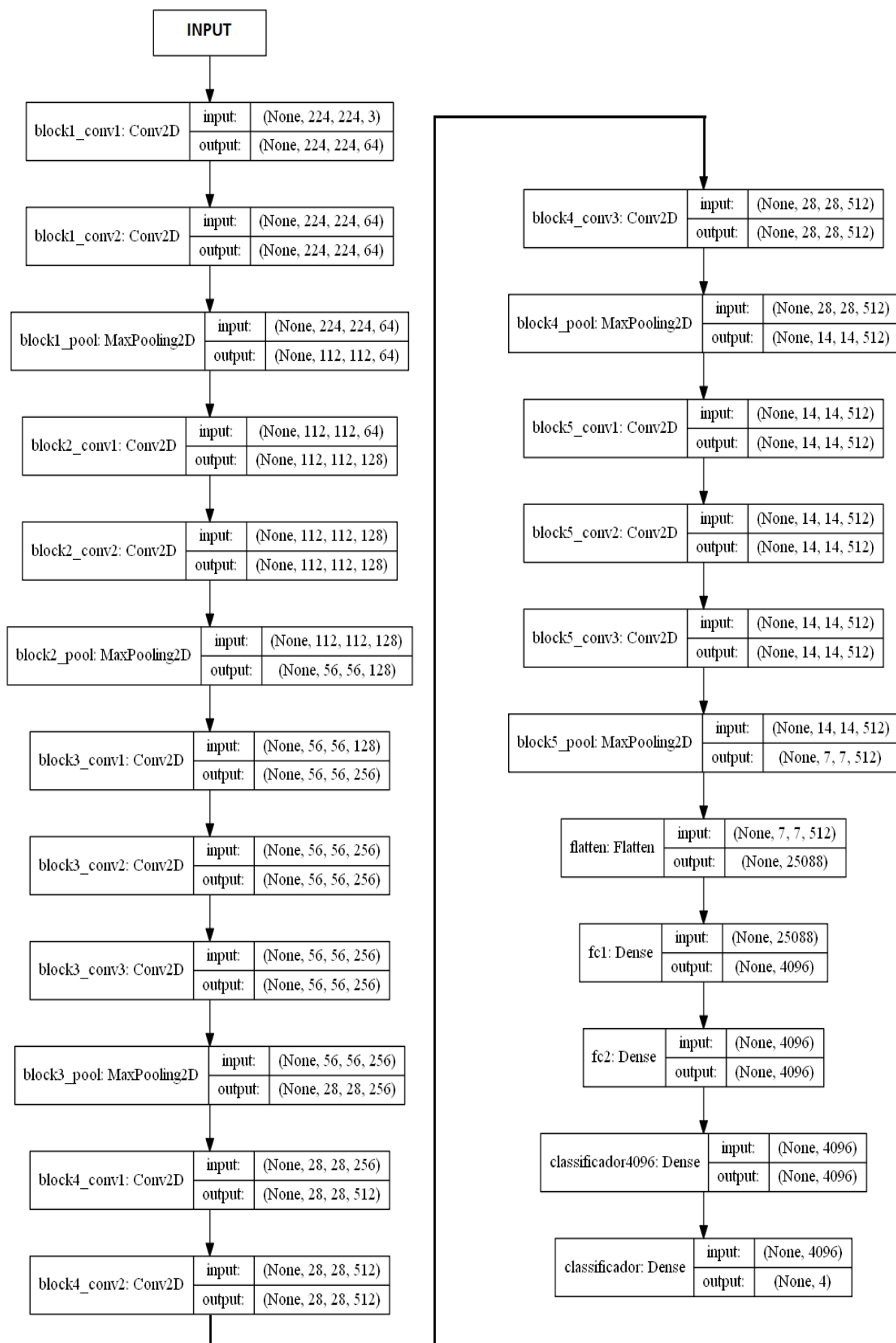


Figura 3.5: Exemplo de Arquitetura com Camadas Classificadoras

## Capítulo 4

# Resultados Experimentais

Neste capítulo, são apresentados os experimentos realizados. Os resultados obtidos pelas arquiteturas escolhidas também são apresentados e discutidos. Primeiramente, são apresentados os experimentos preliminares com um número reduzido de classes. Na sequência serão apresentados os resultados com números maiores de classes. Esses resultados serão tratados em ordem crescente de dificuldade

### 4.1 Identificação de Tatuadores

Os experimentos apresentados nesse capítulo foram executados com o *framework* Keras escrito na linguagem de programação Python. Durante o treinamento, foi aplicado apenas o redimensionamento das imagens para o tamanho da entrada da rede VGG16. Não foi feita nenhuma outra alteração nas imagens. As classes foram apresentadas à rede conforme à sua ordem de criação no banco de dados. Os quatro primeiros artistas a serem incluídos foram de estilos diferentes. Os artistas seguintes são de estilos coincidentes com pelo menos um dos quatro primeiros. A sequência da inclusão dos estilos se encontra numerada abaixo.

1. Krish Trece
2. Matt Beckerich
3. Gabriela Arzabe
4. Pablo Ortiz
5. Mike Rudenball
6. Manu Cruz
7. Dynoz

Os primeiros experimentos conduzidos visavam fazer testes preliminares da rede. Esses experimentos também foram utilizados para encontrar os valores adequados para os hiper-parâmetros



da rede. O primeiro experimento a ser feito foi com apenas duas classes, Krish Trece e Matt Beckerich. Esse experimento foi utilizado para se obter o número de épocas necessárias para o devido treinamento da rede neural. O número encontrado foi de 10 épocas para que todos os modelos. A partir da décima época a melhora do desempenho é muito reduzida, porém o número de épocas foi mantido em vinte. Era esperado que com o aumento do número de classes haveria um aumento da dificuldade da tarefa de identificação. Por consequência, o modelo necessitaria de um número maior de épocas para aprender a executar a tarefa.

Os ensaios contendo até quatro classes foram utilizados para configurar os hiper-parâmetros da rede. O principal ajuste feito foi no número de épocas utilizadas durante o treinamento. O treinamento ocorreu com um número variável de *batches*, uma vez que o número de exemplos aumentava com a inclusão de cada classe. O treinamento foi feito visando aumentar a acurácia de treinamento e reduzir o erro do treinamento. Notou-se que o aprendizado da rede era expressivo até a décima época, mas após este ponto o nível de aprendizado caía consideravelmente. Mantemos um treinamento com 20 épocas, considerando que o erro de validação continuava diminuindo até a vigésima época. Outro ponto avaliado foi o crescimento do número de épocas com o aumento do número de classes.

Durante a fase de teste das arquiteturas, foram empregadas as métricas de acurácia, precisão, revocação e *F-score*. As métricas foram todas aplicadas na rede ao final da vigésima época. Os experimentos foram executados dez vezes para cada combinação de número de classes e arquitetura. Em todas as repetições, foi feito o treinamento de um novo arquivo não havendo o aproveitamento de modelos já treinados. Os resultados apresentados nesse capítulo são as médias das 10 ocorrências dos experimentos. A métrica da acurácia foi aplicada na rede. Ela é utilizada para se ter uma visão geral do desempenho da rede. Como a acurácia sozinha pode levar a conclusões erradas, as métricas precisão, revocação e *F-score* são aplicadas em cada classes separadamente. Elas apresentaram uma visão individual do desempenho de cada classe. Essas métricas possuem valores entre zero e um, mas, para melhor visualização foram passados para a escala de zero a cem. As tabelas, que apresentam as métricas supracitadas, apresentam realce seguindo um código de cores. As células pintadas de vermelho apresentam o melhor desempenho do F-score de uma classe com relação à diferentes arquiteturas utilizadas. De forma similar, a melhor precisão foi colorida com a cor azul e a melhor revocação foi colorida com a cor verde.

Além dessas métricas, foi utilizada a matriz de confusão para representar os resultados da classificação da base de dados de teste. A matriz de confusão apresentada também é a média da matriz de confusão dos dez experimentos executados. A matriz contém em cada linha a classes real e na coluna a classe que foi predita pelo modelo. O valor no interior de cada célula é o número de imagens atribuídas aquela posição. A escala colorida de cada célula é relativa ao número total de imagens de cada classe. Quanto mais escura foi a célula maior é o número de imagens atribuída aquela posição.

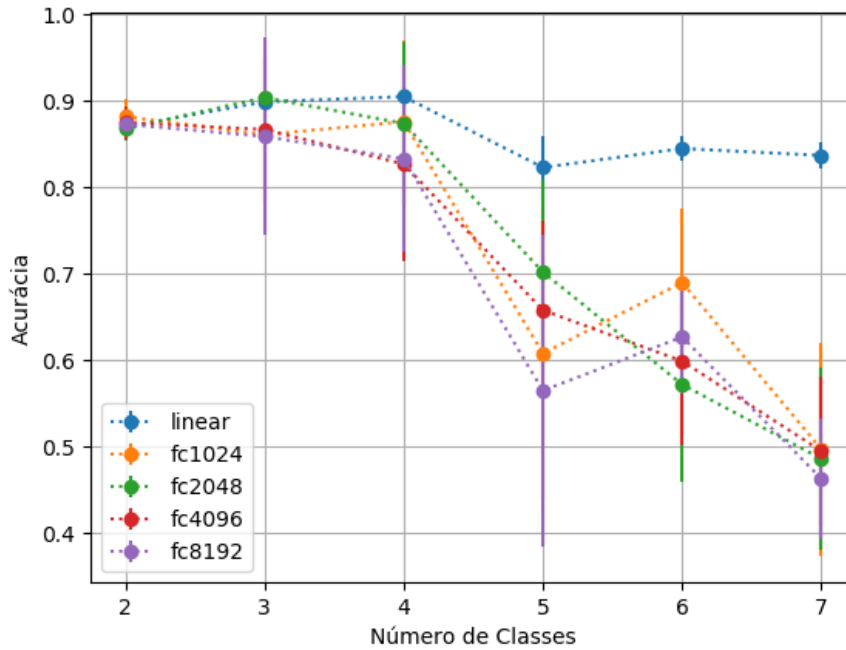


Figura 4.1: Gráfico do desempenho das diferentes arquiteturas a partir do número de classes.

Arquiteturas		2 Classes	3 Classes	4 Classes	5 Classes	6 classes	7 Classes
Linear	Média	0.870	0.898	0.905	0.823	0.845	0.837
	Desvio	0.015	0.006	0.011	0.036	0.014	0.015
FC1024	Média	0.881	0.860	0.875	0.608	0.689	0.497
	Desvio	0.020	0.109	0.093	0.168	0.084	0.124
FC2048	Média	0.868	0.904	0.873	0.701	0.572	0.485
	Desvio	0.014	0.007	0.095	0.108	0.111	0.104
FC4096	Média	0.875	0.866	0.827	0.658	0.599	0.495
	Desvio	0.019	0.083	0.112	0.103	0.096	0.086
FC8192	Média	0.872	0.859	0.832	0.565	0.627	0.462
	Desvio	0.011	0.114	0.107	0.180	0.064	0.069

Tabela 4.1: Tabela contendo os valores da acurácia e o valores desvio padrão dos valores da acurácia para diferentes arquiteturas e diferentes números de classes.

A acurácia da rede está expressa na Figura 4.1 e na Tabela 4.1. Pode se notar que até a inclusão da quinta classe, não há muita variação entre os resultados obtidos. Todas as arquiteturas estão dentro do intervalo de confiança do outro. A partir da inclusão da quinta classe, começa a existir um espaçamento entre os resultados. Em todos os experimentos, a arquitetura Linear se mostrou superior às demais arquiteturas. Esperava-se que houvesse uma tendência entre o número de parâmetros utilizados em cada rede e o seu desempenho. Esse comportamento não foi identificado na medida de acurácia. As outras arquiteturas se alternaram quanto a sua colocação.

### 4.1.1 Experimento com Cinco Classes

A inclusão da quinta classe trouxe consigo a primeira dupla de classes de mesmo estilo. Havendo assim uma relação entre as imagens das classes dos artistas Matt Beckerich e Mike Ridenball. Nesse ponto, esperava-se que houvesse um rendimento inferior nas duas classes em comparação com as outras classes. Olhando para a Tabela 4.2, é possível notar que as duas classes do estilo Irezumi possuem as menores métricas. O mesmo resultado também é comprovado pelas Figuras 4.2, 4.3, 4.4, 4.5 e 4.6 contendo as matrizes de confusão. As duas classes que apresentam grande confusão são as classes supracitadas. Outro ponto que é importante notar é que o erro de classificação não é simétrico. O erro ocorre atribuindo imagens das classes Mike Rudenball como sendo imagens da classe Matt Beckerich.

A confusão assimétrica está presente em todas as arquiteturas com exceção da arquitetura linear. As métricas de qualidade das classes em questão são mais equilibradas na arquitetura Linear. A confusão assimétrica reduz a revocação da quinta classe diminuindo o número de verdadeiros positivos e aumentando o número de falsos negativos. Como há mais Falsos Negativos da quinta classe, há um aumento do número de falsos positivos da segunda classe e por consequência um decréscimo da sua precisão.

Pesquisando um pouco sobre esses dois artistas foi descoberto que ambos são artistas americanos e que o artista Matt Beckerich possui mais tempo de carreira. Pode-se assumir que ele foi uma influência para os trabalhos do quinto artista. Por último, é possível observar que a arquitetura Linear apresentou o melhor resultado, com desempenho superior às demais arquiteturas em quase todas as métricas de quase todas as classes.

		Krish Trece	Matt Beckerich	Gabriela Arzabe	Pablo Ortiz	Mike Rudenball
Linear	F-score	78	68	92	93	76
	Precisão	87	78	93	93	76
	Revocação	74	67	91	94	81
FC1024	F-score	38	54	63	81	41
	Precisão	43	47	65	74	47
	Revocação	43	75	61	93	42
FC2048	F-score	60	59	55	87	37
	Precisão	63	48	57	85	43
	Revocação	67	86	54	93	35
FC4096	F-score	50	59	68	82	38
	Precisão	56	46	70	78	43
	Revocação	53	78	66	92	34
FC8192	F-score	57	55	29	73	23
	Precisão	52	43	30	68	26
	Revocação	69	85	29	84	21

Tabela 4.2: Tabela contendo as médias das métricas de precisão, revocação e F-score para as arquiteturas propostas no Experimento com Cinco Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação.

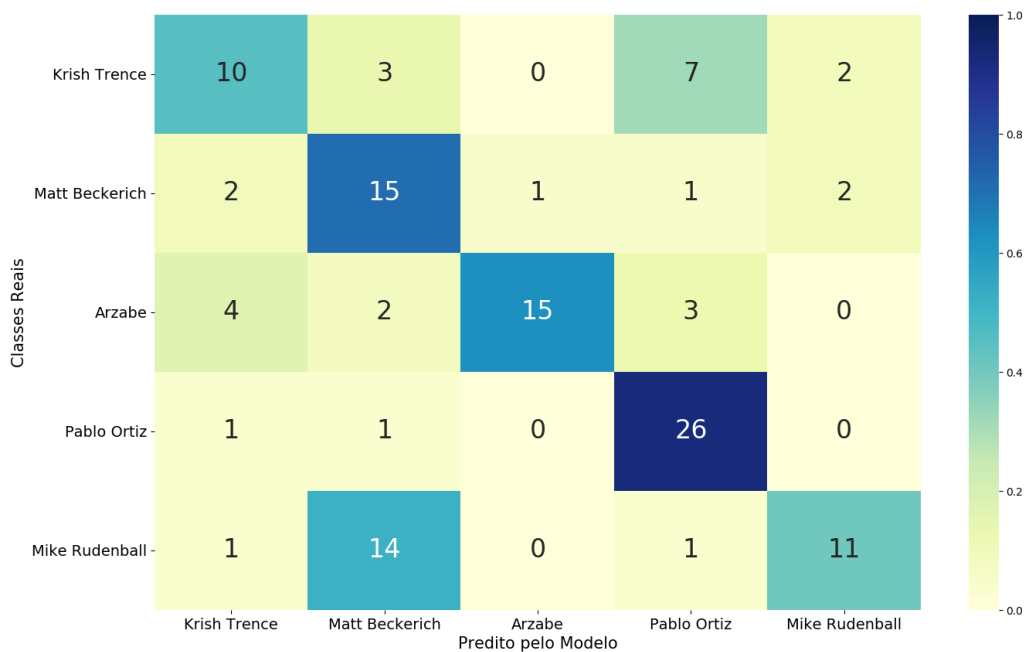


Figura 4.2: Matriz de Confusão da arquitetura FC1024 para o caso de classificação de cinco classes.

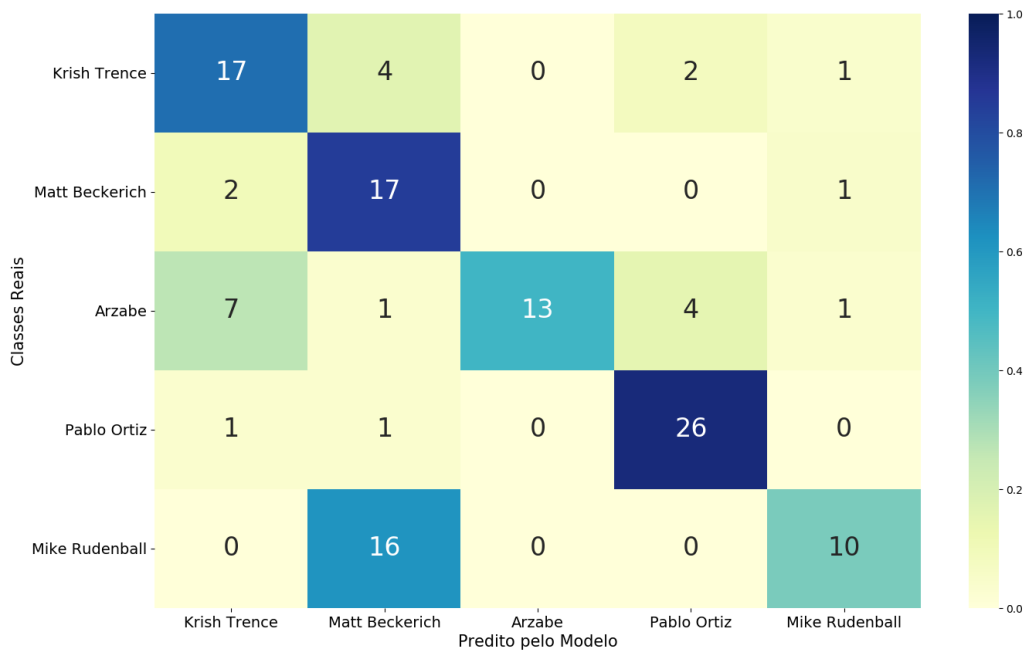


Figura 4.3: Matriz de Confusão da arquitetura FC2048 para o caso de classificação de cinco classes.

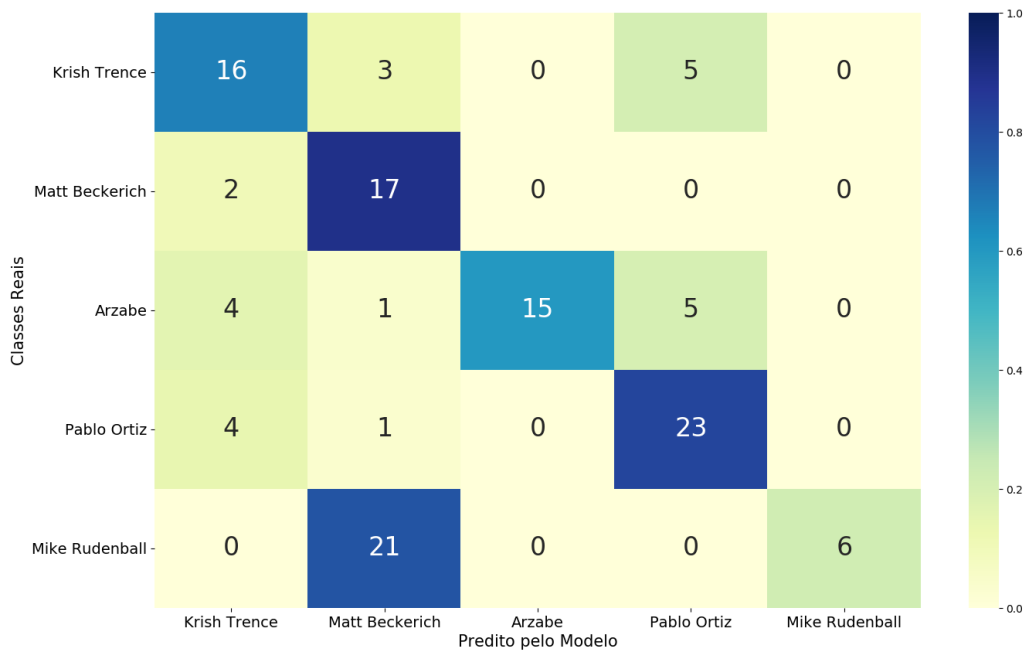


Figura 4.4: Matriz de Confusão da arquitetura FC4096 para o caso de classificação de cinco classes.

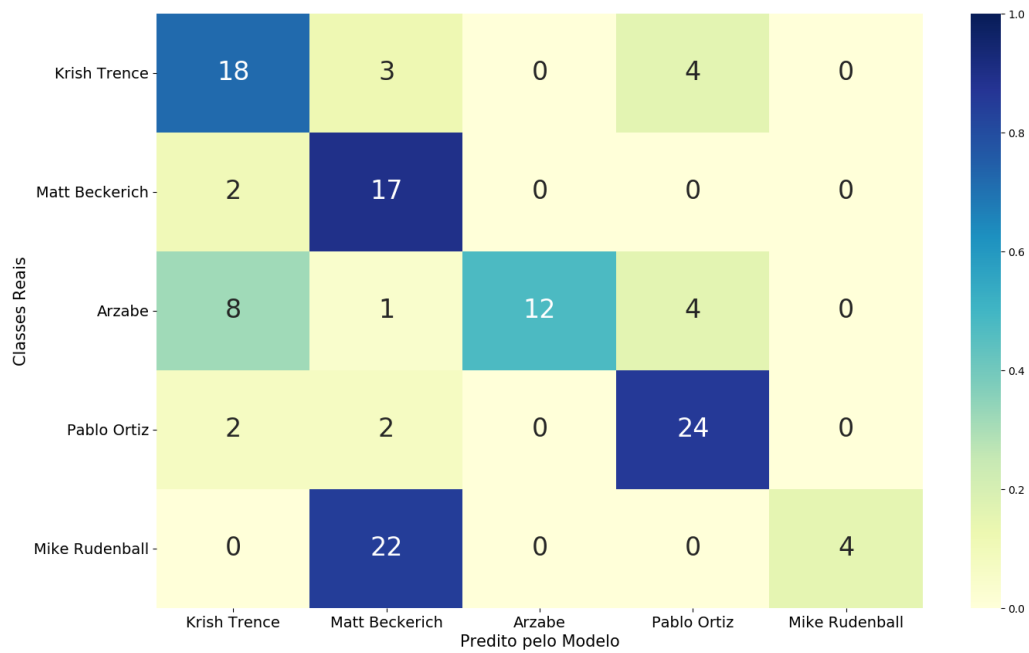


Figura 4.5: Matriz de Confusão da arquitetura FC8192 para o caso de classificação de cinco classes.

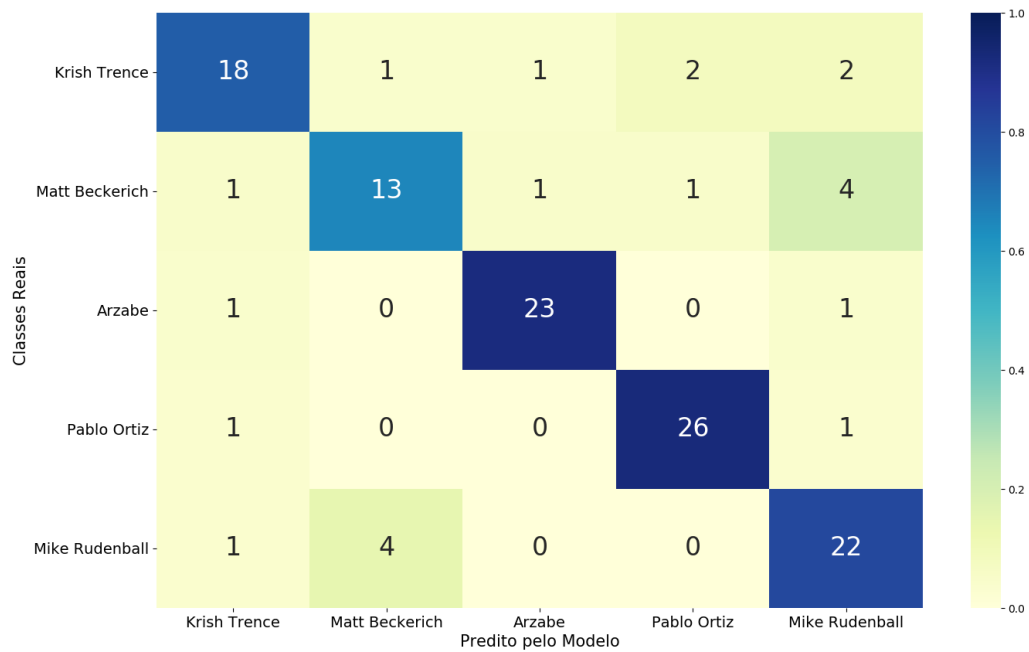


Figura 4.6: Matriz de Confusão da arquitetura Linear para o caso de classificação de cinco classes.

### 4.1.2 Experimento com Seis classes

Neste experimento foi incluída a sexta classe, Manu Cruz. Ele possui o mesmo estilo da artista Krish Trece, além disso ambos trabalham no mesmo estúdio. Era esperado que houvesse uma grande correlação entre os dois artistas.

Olhando para o desempenho da arquitetura Linear na Tabela 4.3, é possível notar que há um aumento generalizado dos indicadores de desempenho. A inclusão da sexta classe não só beneficiou a classificação do seu par, mas também as classes Matt e Mike. Atribui-se a melhora do desempenho a um aumento das características extraídas. As características introduzidas pela sexta classe beneficiaram o ajuste da função classificadora de diferentes classes.

No entanto, houve um agravamento da confusão entre classes. As quatro primeiras classes apresentaram melhor desempenho com a introdução da sexta classe. A quinta e a sexta classe tiveram resultados piores. A confusão dessas classes se fortaleceu com o aumento do número de parâmetros da rede. As arquiteturas FC4096 e FC8192 apresentaram desempenhos muito baixos para a classificação dessas classes. Olhando as matrizes de confusão, é possível ver que, com exceção da arquitetura Linear, todas as outras arquiteturas não classificaram as duas últimas classes mais vezes como o seu par de estilo. O aumento do número de parâmetros tende a privilegiar as classes introduzidas anteriormente apesar da rede ser retreinada em cada um dos experimentos. É possível notar que a rede dá preferência às classes introduzidas em experimentos passados. Esse resultado não esperado, uma vez que a apresentação das imagens é feita em ordem aleatória e as classes possuem números semelhantes de exemplos. Por fim, a arquitetura Linear se mostrou novamente superior às outras. Como pode ser visto na Tabela 4.3 o seu desempenho foi superior na métrica F-score em todas as classes.

		Krish Trece	Matt Beckerich	Gabriela Arzabe	Pablo Ortiz	Mike Rudenball	Manu Cruz
Linear	F-score	81	71	98	96	75	84
	Precisão	81	64	100	97	83	86
	Revocação	82	82	96	94	69	83
FC1024	F-score	68	63	78	90	38	34
	Precisão	57	53	80	89	42	34
	Revocação	88	84	77	94	35	34
FC2048	F-score	48	57	58	72	30	17
	Precisão	46	45	58	66	33	16
	Revocação	67	87	58	83	27	19
FC4096	F-score	59	57	68	84	22	18
	Precisão	51	44	68	79	25	18
	Revocação	79	86	67	93	20	18
FC8192	F-score	57	59	77	87	28	8
	Precisão	48	47	79	84	33	7
	Revocação	82	76	75	92	25	9

Tabela 4.3: Tabela contendo as médias das métricas de Precisão, Revocação e F-score para as arquiteturas propostas no Experimento com Seis Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação.

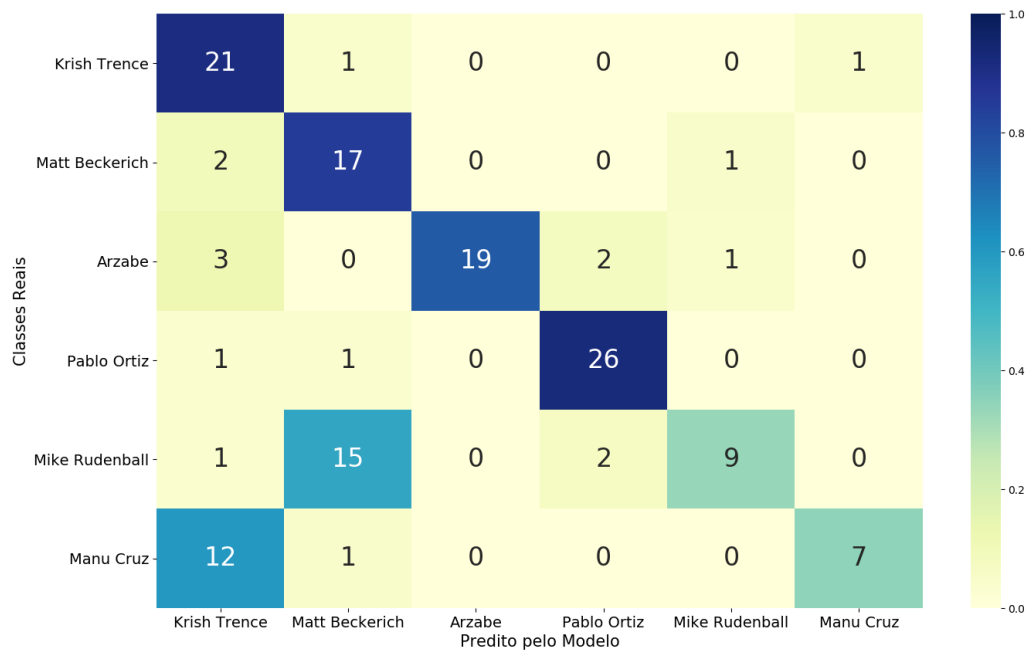


Figura 4.7: Matriz de Confusão da arquitetura FC1024 para o caso de classificação de seis classes.

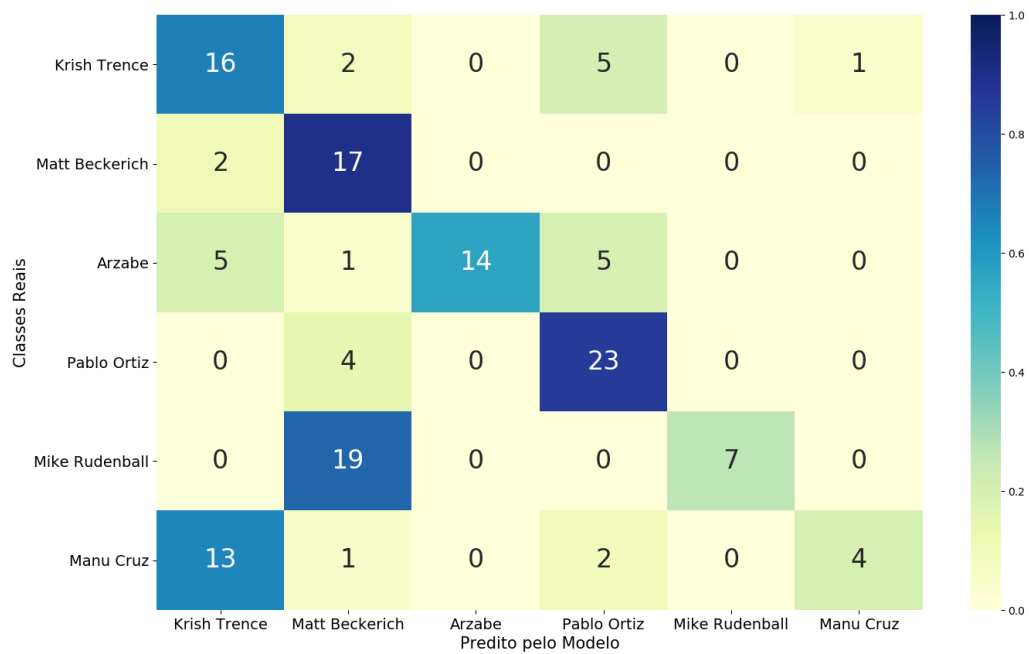


Figura 4.8: Matriz de Confusão da arquitetura FC2048 para o caso de classificação de seis classes.



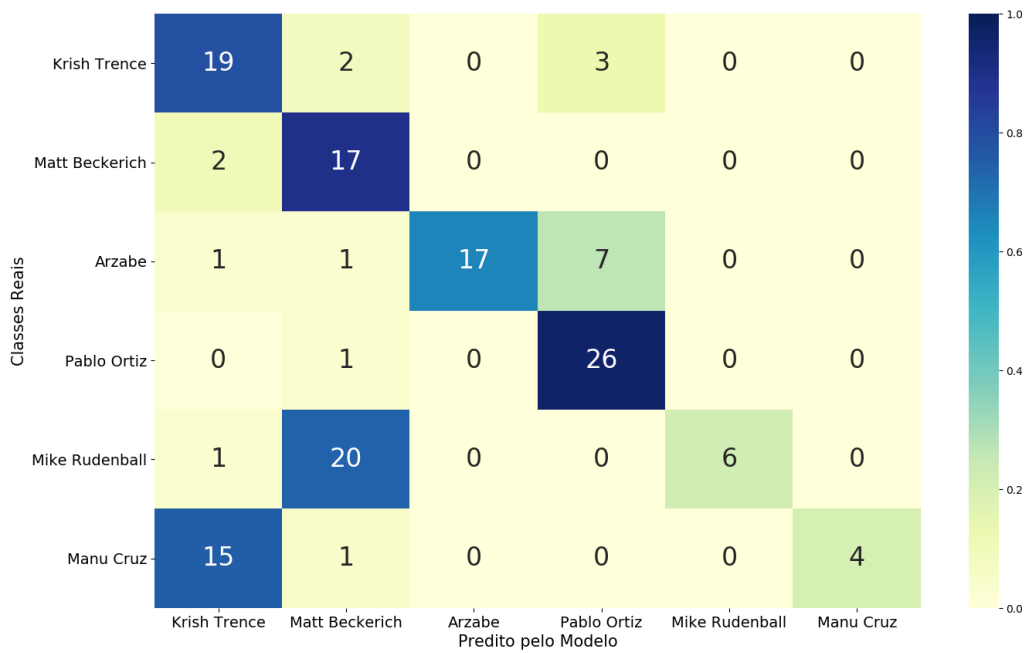


Figura 4.9: Matriz de Confusão da arquitetura FC4096 para o caso de classificação de seis classes.

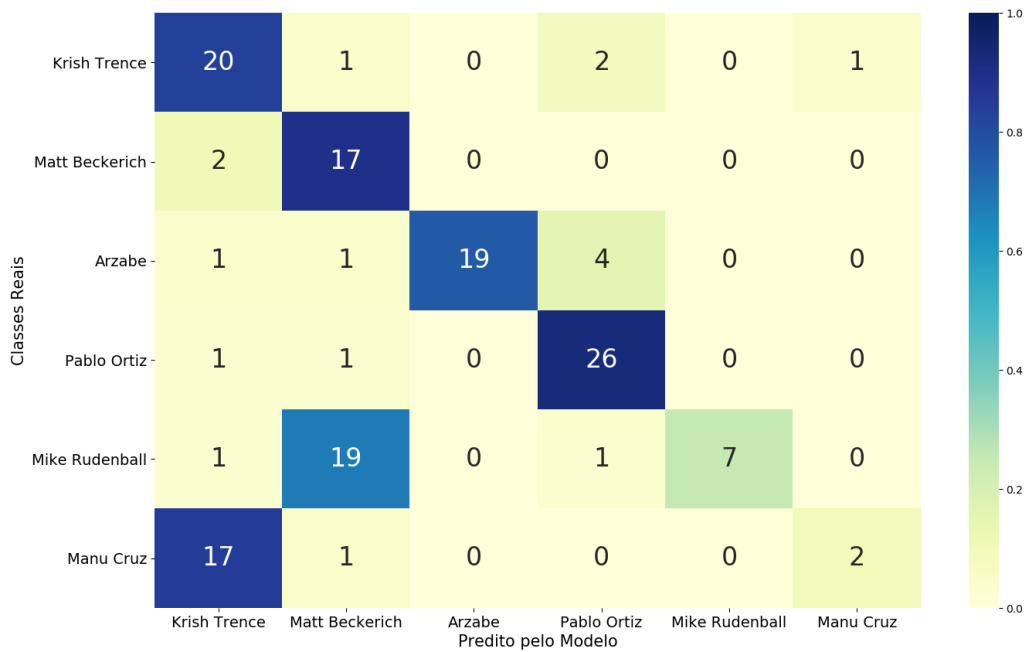


Figura 4.10: Matriz de Confusão da arquitetura FC8192 para o caso de classificação de seis classes.

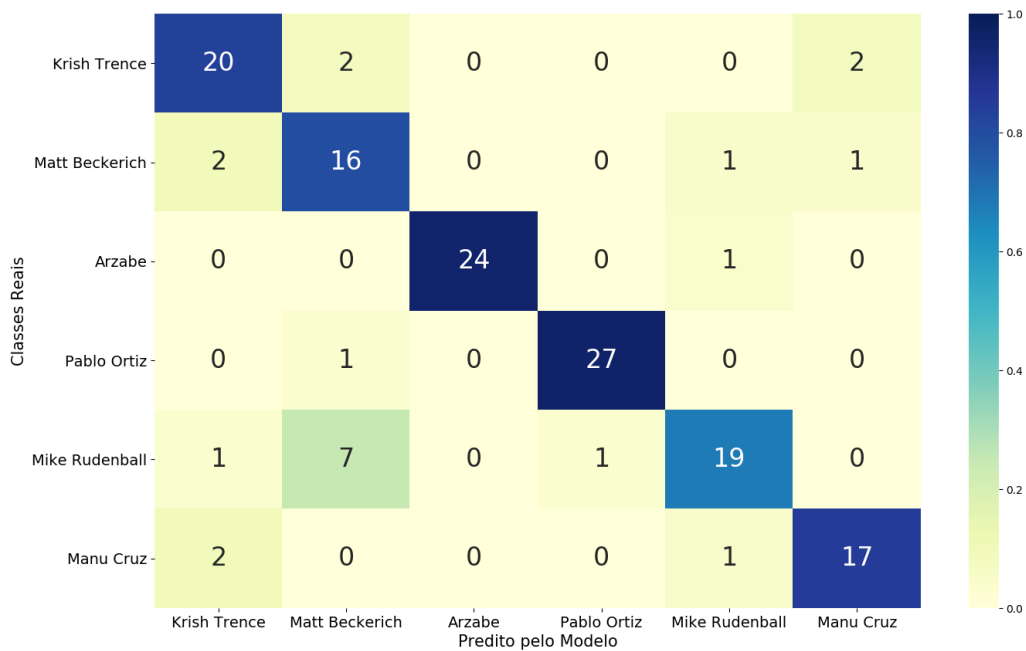


Figura 4.11: Matriz de Confusão da arquitetura Linear para o caso de classificação de seis classes.

### 4.1.3 Experimento com Sete Classes

A introdução da sétima classe, Dynoz, criou o terceiro par de artistas de mesmo estilo. Com isso, era esperado uma queda da métrica da classe Pablo Ortiz, também do estilo *Water color*. A queda foi verificada, porém não foi grande. A precisão e o F-score de ambos os artistas foram superiores a 90, sendo superados apenas pela classe Arzabe que não possui um par de mesmo estilo. Diferente dos outros experimentos, esse par de classes foi o primeiro a não apresentar confusão assimétrica nas matrizes de confusão. É possível este comportamento na Tabela 4.4. As classes Dynoz e Pablo Ortiz apresentam decaimento em patamares similares com o aumento do número de parâmetros.

Novamente há um decaimento geral do valor das métricas à medida que se aumenta o número de parâmetros da camada interna do classificador. Nesse experimento, a queda foi mais abrupta do que nos experimentos anteriores. Inclusive em classes como Arzabe e Pablo Ortiz que não apresentaram quedas grandes em experimentos anteriores. A quinta e sexta classes apresentaram as piores métricas possíveis.

Houve um espalhamento geral das predições erradas de todas as classes. As Figuras 4.12, 4.13, 4.14, 4.15 e 4.16 apresentam esses resultados. Na arquitetura FC8192, o erro de todas as imagens de duas classes como já citado. As arquiteturas com camadas ocultas de maneira geral foram incapazes de descrever corretamente a separação entre as duplas de classes em todos os experimentos. A confusão assimétrica se mostrou presente nos três experimentos. As arquiteturas com camadas

internas se mostraram ineficientes na separação dessas classes. Como há uma assimetria entre as classificações é possível concluir que a fronteira está entrando mais no espaço de uma das classes do que no espaço da outra.

A arquitetura Linear se mostrou superior às demais na classificação dos artistas em todos os experimentos realizados. Conforme dito no Capítulo 3, as camadas da VGG16 estão trabalhando como um extrator de características. O aprendizado dessas camadas foi bloqueado e elas não atualizam seus parâmetros. Sendo assim, o melhor desempenho deve ser totalmente atribuído à parte classificadora da rede. Comparando as diferentes arquiteturas, é possível notar que o desempenho da rede está ligado ao número de parâmetros das camadas classificadora.

		Krish Trece	Matt Beckerich	Gabriela Arzabe	Pablo Ortiz	Mike Rudenball	Manu Cruz	Dynoz
Linear	F-score	82	68	98	90	75	84	90
	Precisão	83	58	100	96	85	76	91
	Revocação	82	86	96	85	67	84	89
fc1024	F-score	51	51	37	56	8	8	51
	Precisão	42	37	37	53	9	8	60
	Revocação	88	78	37	64	7	9	60
fc2048	F-score	56	52	58	32	8	8	59
	Precisão	42	33	60	34	7	7	49
	Revocação	87	68	56	31	9	9	79
fc4096	F-score	44	54	39	52	21	0	72
	Precisão	32	41	40	51	28	0	68
	Revocação	72	78	38	56	20	0	82
fc8192	F-score	56	49	39	42	0	0	57
	Precisão	42	34	40	35	0	0	54
	Revocação	86	88	38	54	0	0	65

Tabela 4.4: Tabela contendo as médias das métricas de Precisão, Revocação e F-score para as arquiteturas propostas no Experimento com Sete Classes. Realce na cor vermelha para a arquitetura com o melhor F-score. Realce na cor azul para a melhor precisão. Realce na cor verde para a melhor Revocação.

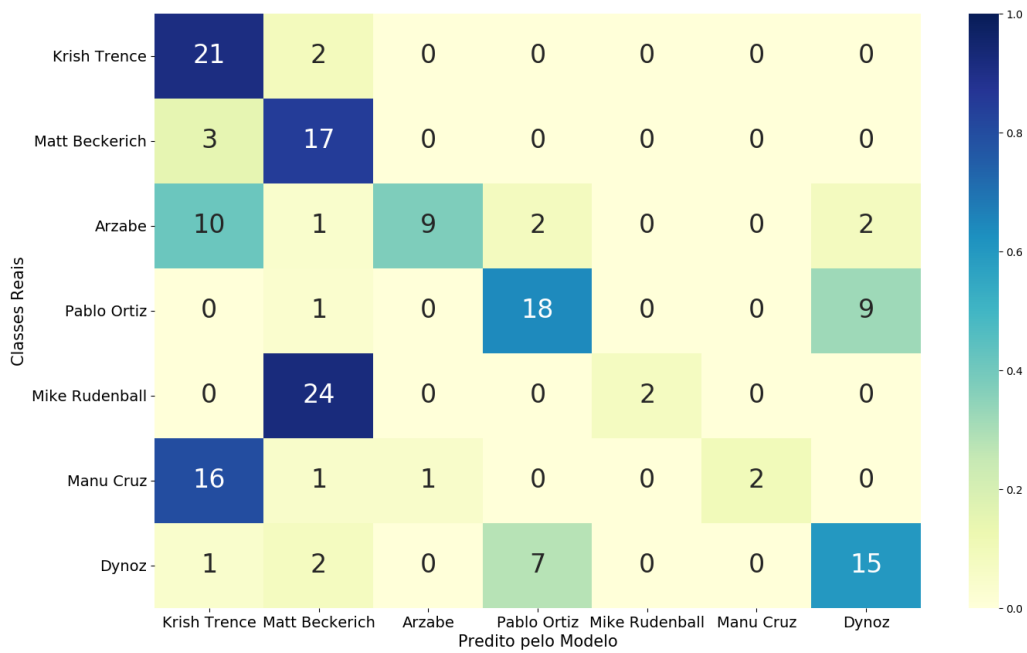


Figura 4.12: Matriz de Confusão da arquitetura FC1024 para o caso de classificação de sete classes.

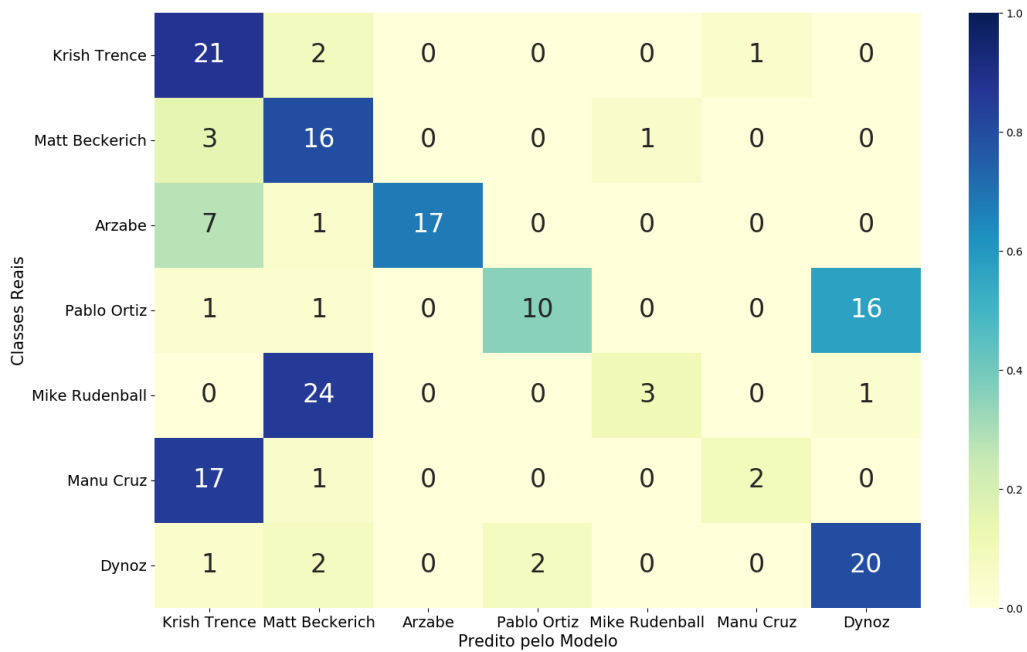


Figura 4.13: Matriz de Confusão da arquitetura FC2048 para o caso de classificação de sete classes.

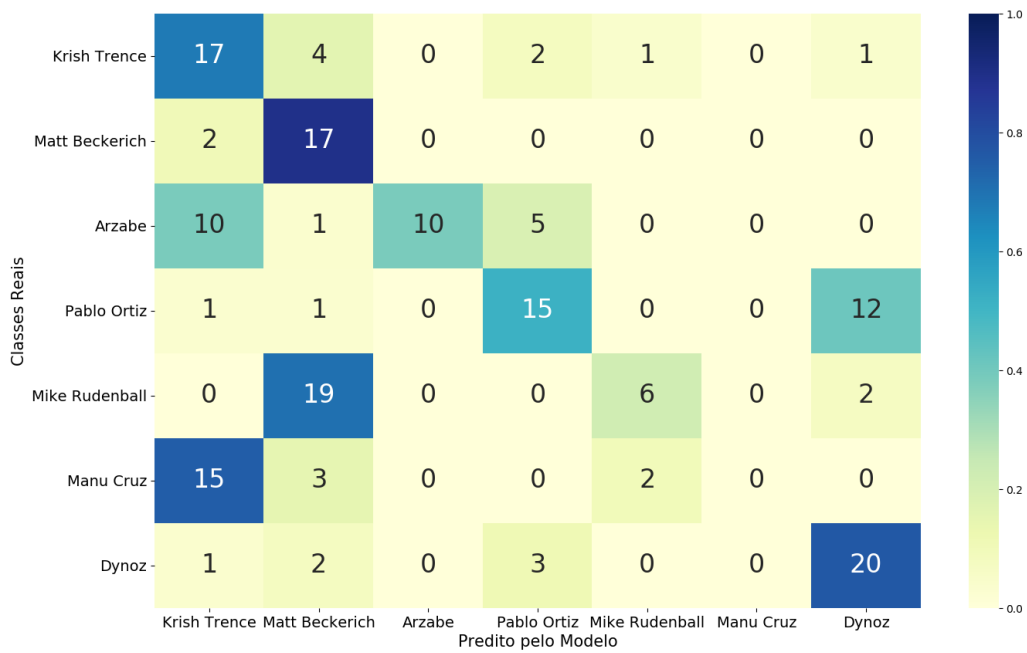


Figura 4.14: Matriz de Confusão da arquitetura FC4096 para o caso de classificação de sete classes.

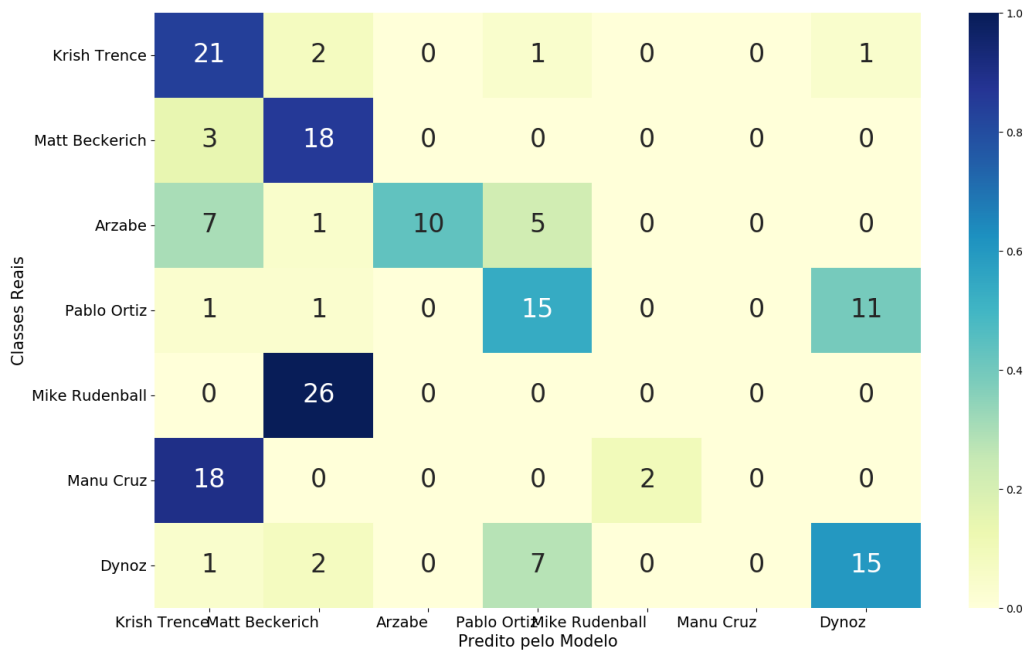


Figura 4.15: Matriz de Confusão da arquitetura FC8192 para o caso de classificação de sete classes.

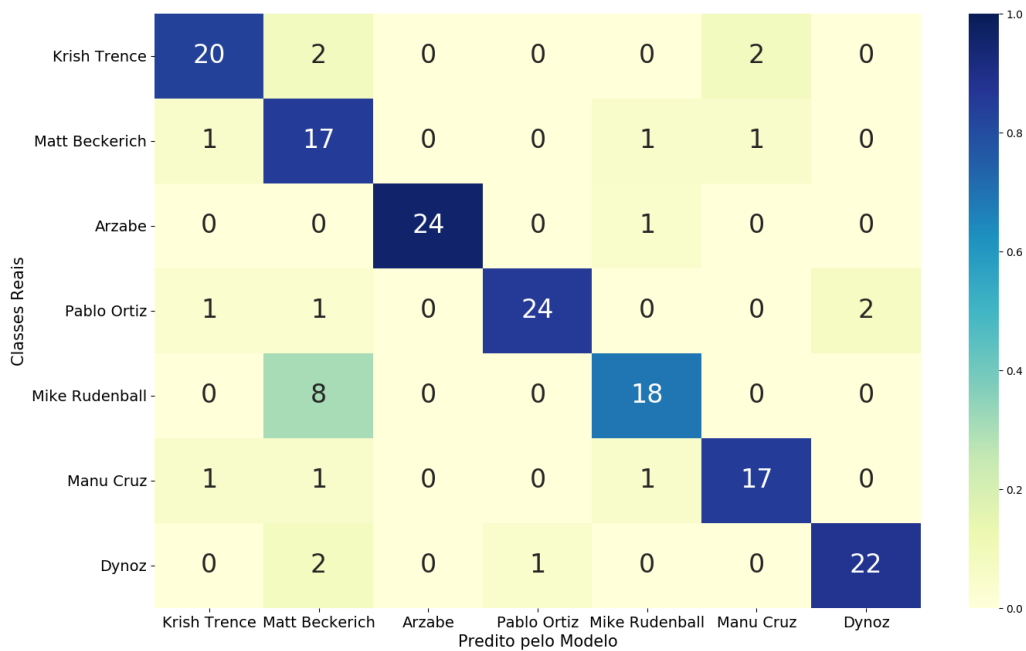


Figura 4.16: Matriz de Confusão da arquitetura Linear para o caso de classificação de sete classes.

#### 4.1.4 Discussão dos Resultados

Comparando os resultados presentes nas Tabelas 4.2, 4.3 e 4.4, tem-se que o desempenho da arquitetura é inversamente proporcional ao número de parâmetros. Esse resultado ocorre devido ao banco de dados reduzido. O número de imagens de cada classe não é suficiente para a o devido treinamento do número de parâmetros das redes classificadoras com camadas internas.

Os resultados dos experimentos anteriores mostram que a arquitetura Linear tem o melhor desempenho dentre as arquiteturas propostas. A consistência dos resultados necessita de uma análise individualizada. Segue a análise da razão da arquitetura Linear ter resultado superior as demais.

Conforme explicado no capítulo de 3, a rede VGG16 foi modificada e utilizada como o extrator de características desse algoritmo. A mesma arquitetura da VGG16 foi utilizada em todas as diferentes arquiteturas propostas nessa monografia. Os valores dos pesos das camadas da VGG16 foram os disponibilizados pelo Keras. As camadas foram congeladas durante o treinamento. Logo, os pesos não foram atualizados durante o treinamento. Conclui-se que o desempenho se dá pelas camadas classificadoras.

Arquitetura	5 classes	6 classes	7 classes
Linear	20.485	24.582	28.679
FC1024	4.200.453	4.201.478	4.202.503
FC2048	8.400.901	8.402.950	8.404.999
FC4096	16.801.797	16.805.894	16.809.991
FC8192	33.603.589	33.611.782	33.619.975

Tabela 4.5: Tabela com os número de parâmetros atualizáveis de cada arquitetura.

Acredita-se que o desempenho superior da arquitetura sem camadas internas se deve ao fato do seu número reduzido de parâmetros. O número de parâmetros de cada arquitetura está listado na Tabela 4.5. Nota-se que o número de parâmetros cresce rapidamente com o aumento do número de neurônios. A quantidade de parâmetros não seria um problema se houvesse um número equivalente de imagens na base de dados. Além disso, o número de parâmetros é demasiadamente grande. Quando comparado com outros modelos, é possível ver que se pode diminuir o número de parâmetros. A Tabela 4.6 apresenta outras arquiteturas de rede convolucionais e seus respectivos números de parâmetros e quantidades de camadas.

Modelo	Número de parâmetros	Profundidade
Xception	22.910.480	126
VGG16	138.357.544	23
VGG19	143.667.240	26
ResNet50	25.636.712	168
InceptionV3	23.851.784	159
InceptionResNetV2	55.873.736	572
MobileNet	4.253.864	88
MobileNetV2	3.538.984	88
DenseNet121	8.062.504	121
DenseNet169	14.307.880	169
DenseNet201	20.242.984	201

Tabela 4.6: Tabela contendo o número de parâmetros de diferentes modelo de rede convolucionais utilizadas na classificação de imagens.

# Capítulo 5

## Conclusões

O objetivo desse trabalho era criar um algoritmo capaz de identificar a autoria de tatuagens. Sabendo que rede neurais convolucionais já haviam sido utilizadas para a identificação da autoria de obras de arte [7, 8] e para o reconhecimento de tatuagens [35, 36, 37, 38]. Fez-se o uso desse algoritmo.

Devido a aplicação proposta por esse trabalho, não existia um banco de dados de imagens de tatuagens que pudesse ser utilizado. Foi necessário criar um bando de dados personalizado. Utilizou-se a rede social Instagram para a aquisição das fotos. Foi feito um processo de seleção somente das imagens que possuísem tatuagens. As imagens selecionadas passaram por um processo de higienização. A higienização visou eliminar padrões visuais do ambiente do estúdio que pudessem interferir na classificação. A criação do banco de dados é uma contribuição para trabalhos futuros nessa área.

Em posse do banco de dados, notou-se que o número de imagens disponíveis para treinamento ainda era pequeno se comparado aos números empregados em rede neurais convolucionais. Buscou-se alternativas para contornar esse problema. Escolheu-se a rede VGG16 treinadas na base de dados da Imagenet. Foi necessário fazer uma modificação na transferência de aprendizado devido à escassez de imagens. Ao invés de permitir que as últimas camadas convolucionais pudessem ser treinadas, bloqueou-se totalmente o treinamento e utilizou-se a VGG16 como um extrator de características. Adicionaram-se camadas densas ao final da VGG16 para realizar a classificação das imagens. Essa estrutura se mostrou competente na identificação dos tatuadores.

Foram criadas cinco variedades de arquiteturas de rede neurais convolucionais. Uma com apenas uma camada classificadora e as outras quatro com diferentes tamanhos de camadas internas. Foram realizados experimentos com um crescente número de classes. As arquiteturas apresentaram desempenho similar até a inclusão da quinta classes onde as arquiteturas começaram a divergir quanto ao desempenho. A arquitetura mais simples apresentou o melhor desempenho dentre todas, identificando corretamente 83,7% das imagens. As outras arquiteturas apresentaram uma relação entre o número de parâmetros treináveis da rede e o seu desempenho. Sendo que a arquitetura com a maior camada interna apresentou o pior desempenho.

Notou-se que para as arquiteturas com camadas internas houve uma relação de erro de classi-



ficação não simétrica, havendo um maior erro de uma das classes que apresentavam semelhanças. Concluiu-se que a diferença é oriunda das camadas classificadoras já que as camadas da VGG16 apresentavam os mesmos pesos em todos os cenários. As arquiteturas com camadas internas apresentaram um número muito superior do que a arquitetura sem camadas internas. Conclui-se que houve um desbalanço entre a quantidade de parâmetros e de imagens disponíveis para treinamento. Acredita-se que um número maior de imagens disponibilizaria mais características para o ajuste dos parâmetros das camadas classificadoras.

Para trabalhos futuros, pode-se testar novas arquiteturas. Como dito anteriormente, foi necessário fazer muitas simplificações na VGG16 para que o desempenho do modelo pudesse melhorar. Caso fosse possível aumentar o número de imagens de cada classe, poderia se fazer alterações no número de camadas treináveis. Outra possibilidade é a utilização de outras arquiteturas na transferência de aprendizado. A arquitetura escolhida foi uma arquitetura sequencial, ou seja, há um único caminho no qual a informação passa. Trabalhos mais recentes sugerem a implementação de caminhos em paralelo [39, 40, 41]. Os resultados dessas redes no desafio Imagenet foram superiores ao da VGG16, sugerindo uma melhora do desempenho da identificação. Outros trabalhos propõem ligações com saltos de camadas [42, 43]. Essas ligações com saltos fariam com que as camadas mais profundas da rede recebessem dados das camadas não sequencialmente ligadas a elas. Aumentando a informação disponível para cada camada da rede sem aumentar o número de imagens do banco de dados. É válido lembrar que tanto as redes com múltiplos caminhos tanto as com ligações em saltos apresentam menos parâmetros do que a VGG16. Indicando pelos resultados analisados desse trabalho que os resultados dessas da identificação da autoria de tatuagens já seria melhor do que o da VGG16.

Outro ponto não trabalhado que pode ser desenvolvido são técnicas de visualização de rede convolucionais. A rede neural utilizadas neste trabalho foi tratada como uma caixa preta, não se sabe quais partes da imagem são considerada para realizar o aprendizado. Técnicas de visualização dos filtros das camadas convolucionais já foram criadas [44]. Existem diferentes técnicas de impressão das regiões de maior importância das imagens para a classificação. Dentre essas técnicas temos Mapas de Calor e Mapas de Saliência [45, 46]. Essas técnicas seriam importantes para saber se a higienização do banco de dados foi satisfatória. Caso essas técnicas mostrassem que a rede está utilizando regiões não relacionadas as tatuagens, seria necessário reestruturar os procedimentos realizados na criação do banco de dados e no treinamento da rede.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] GARDNER, E. *Judge Refuses to Dismiss Copyright Lawsuit Over LeBron James Tattoo in 'NBA 2K'*. [S.l.], Acesso em: 2 mai 2018. Disponível em: <<https://www.hollywoodreporter.com/thr-esq/judge-refuses-dismiss-copyright-lawsuit-tattoos-nba-2k-1098620>>.
- [2] HEITNER, D. *New Tattoo Copyright Infringement Case Filed By Artist Who Inked WWE Wrestler Randy Orton*. [S.l.], Acesso em: 2 mai 2018. Disponível em: <<https://www.forbes.com/sites/darrenheitner/2018/04/18/new-tattoo-copyright-infringement-case-filed-by-artist-who-inked-wwe-wrestler-randy-orton/1a9999c34edb>>.
- [3] KHAN, S. *Ethem Alpaydin. Introduction to Machine Learning (Adaptive Computation and Machine Learning Series)*. Cambridge: The MIT Press, 2004.
- [4] SAMARASINGHE, S. *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. [S.l.]: Auerbach publications, 2016.
- [5] GOODFELLOW, I. et al. *Deep learning*. Cambridge: The MIT Press, 2016.
- [6] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.
- [7] NOORD, N. V.; HENDRIKS, E.; POSTMA, E. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, IEEE, v. 32, n. 4, p. 46–54, 2015.
- [8] VISWANATHAN, N. Artist identification with convolutional neural networks. *transfer*, v. 77, p. 89–8.
- [9] LOMBARDI, T. E. *Classification of style in fine-art painting*. Citeseer, 2005.
- [10] LECOUTRE, A.; NEGREVERGNE, B.; YGER, F. Recognizing art style automatically in painting with deep learning. p. 327–342, 2017.
- [11] GONZALEZ, R. C.; WOODS, R. E. et al. *Digital image processing*. Upper Saddle River, New Jersey: Pearson-Prentice-Hall, 2002.
- [12] LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, MIT Press, v. 1, n. 4, p. 541–551, 1989.

- [13] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. Cambridge: The MIT Press, 2012. p. 1097–1105.
- [14] BISHOP, C. M. *Pattern Recognition and Machine Learning*. Salmon Tower Building NY: Springer, 2006.
- [15] TIELEMAN, T.; HINTON, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, v. 4, n. 2, p. 26–31, 2012.
- [16] VALVERDE-ALBACETE, F. J.; PELÁEZ-MORENO, C. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PLoS one*, Public Library of Science, v. 9, n. 1, p. e84217, 2014.
- [17] SØRENSEN, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, v. 5, p. 1–34, 1948.
- [18] DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, Wiley Online Library, v. 26, n. 3, p. 297–302, 1945.
- [19] NGAN, M.; GROTHOR, P. Tattoo recognition technology-challenge (tatt-c): an open tattoo database for developing tattoo recognition research. p. 1–6, 2015.
- [20] BECKERICH, M. *mattbeckerich*. [S.l.], (Acesso em 5 de mai 2018). Disponível em: <<https://www.instagram.com/mattbeckerich/>>.
- [21] RUDENBALL, M. *mikerudeball*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <<https://www.instagram.com/mikerubendall/?hl=pt-br>>.
- [22] TRECE, K. *krishtattoo*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <<https://www.instagram.com/krishtattoo/?hl=pt>>.
- [23] CRUZ, M. *manu\_raccoon*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <[https://www.instagram.com/manu\\_raccoon/](https://www.instagram.com/manu_raccoon/)>.
- [24] ORTIZ, P. *pablo\_ortiz\_tattoo*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <[https://www.instagram.com/pablo\\_ortiz\\_tattoo/?hl=pt-br](https://www.instagram.com/pablo_ortiz_tattoo/?hl=pt-br)>.
- [25] DYNOZ. *dynozartattack*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <<https://www.instagram.com/dynozartattack/?hl=pt-br>>.
- [26] LEHMKUHL, G. A. *gabriela\_arzabe*. [S.l.], Acesso em: 5 mai 2018. Disponível em: <[https://www.instagram.com/gabriela\\_arzabe/?hl=pt-br](https://www.instagram.com/gabriela_arzabe/?hl=pt-br)>.
- [27] PARETO, V.; SCHWIER, A. S. *Manual of political economy Tr. by Ann S. Schwier*. London: Macmillan, 1927.

- [28] BUDA, M.; MAKI, A.; MAZUROWSKI, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, Elsevier, v. 106, p. 249–259, 2018.
- [29] SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [30] DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009.
- [31] OQUAB, M. et al. Learning and transferring mid-level image representations using convolutional neural networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York: IEEE, 2014.
- [32] YOSINSKI, J. et al. How transferable are features in deep neural networks? In: *Advances in neural information processing systems*. Cambridge: The MIT Press, 2014. p. 3320–3328.
- [33] SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
- [34] CARUANA, R.; LAWRENCE, S.; GILES, C. L. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: *Advances in neural information processing systems*. Cambridge: The MIT Press, 2001. p. 402–408.
- [35] XU, Q. et al. Tattoo detection based on cnn and remarks on the nist database. In: *IEEE. Biometrics (ICB), 2016 International Conference on*. New York: IEEE, 2016. p. 1–7.
- [36] SUN, Z. H. et al. Tattoo detection and localization using region-based deep learning. In: *IEEE. Pattern Recognition (ICPR), 2016 23rd International Conference on*. New York: IEEE, 2016. p. 3055–3060.
- [37] DI, X.; PATEL, V. M. Deep tattoo recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. New York: IEEE, 2016. p. 51–58.
- [38] HRKAĆ, T. et al. Deep learning architectures for tattoo detection and de-identification. In: *IEEE. 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*. New York: IEEE, 2016. p. 1–5.
- [39] SZEGEDY, C. et al. Going deeper with convolutions. In: *CVPR*. [S.l.], 2015.
- [40] SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 2818–2826.
- [41] LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- [42] HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. v. 1, n. 2, p. 3.
- [43] HE, K. et al. Deep residual learning for image recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York: IEEE, 2016.

- [44] YOSINSKI, J. et al. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.
- [45] ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 818–833.
- [46] SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.