UMA ABORDAGEM HÍBRIDA COM REDES DE ARQUITETURA PROFUNDA APLICADA AO RECONHECIMENTO DE EXPRESSÕES FACIAIS

Trabalho de Conclusão de Curso
Engenharia da Computação

Nestor Tavares Maciel Júnior
Orientador: Prof. Dr. Bruno José Torres Fernandes

UNIVERSIDADE
DE PERNAMBUCO

Nestor Tavares Maciel Júnior

UMA ABORDAGEM HÍBRIDA COM REDES DE ARQUITETURA PROFUNDA APLICADA AO RECONHECIMENTO DE EXPRESSÕES FACIAIS

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco

Universidade de Pernambuco

Escola Politécnica de Pernambuco

Graduação em Engenharia de Computação

Orientador: Prof. Dr. Bruno José Torres Fernandes

Recife - PE, Brasil

3 de dezembro de 2015

ESCOLA POLITÉCNICA
DE PERNAMBUCO

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 2 de 12 de 2015, às 9:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente NESTOR TAVARES MACIEL JUNIOR, orientado pelo professor Bruno José Torres Fernandes, sob título UMA ABORDAGEM HÍBRIDA COM REDES DE ARQUITETURA PROFUNDA APLICADA AO RECONHECIMENTO DE EXPRESSÕES FACIAIS, a banca composta pelos professores:

Sérgio Murilo Maciel Fernandes

Bruno José Torres Fernandes

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 10,0 (dez)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 01 dias para entrega da versão final da monografia a contar da data deste documento.

SÉRGIO MURILO MACIEL FERNANDES

BRUNO JOSÉ TORRES FERNANDES

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

À minha família.

Agradecimentos

Agradeço, primeiramente, a Deus. Agradeço-O por todas as oportunidades, cuidado, e motivação em momentos de dificuldade. Se tive forças de chegar até aqui, foi porque Ele esteve ao meu lado.

Agradeço, também, à minha família. Meus pais, Nestor e Sandra, sempre presentes e dedicados, fornecendo apoio, carinho, motivação, amor e conselhos. Às minhas irmãs, Suellen e Susanne, obrigado por toda a paciência e inspiração.

Agradeço, ainda, aos grandes amigos(as) que fiz nesta jornada: Arthur, Felipe, Gearlles e Milla. Mais de cinco anos de muitas risadas, estudos, projetos, sufocos e diversão. Vocês fazem parte da minha formação não só como engenheiro da computação, mas como indivíduo.

Agradeço também aos meus mentores durante esta trajetória, por todo o acompanhamento, conselhos e confiança. Prof. Bruno, prof. Byron e prof. Sérgio Murilo, vocês foram peças fundamentais da minha formação.

Agradeço, por fim, a todos os que colaboraram em minha formação, seja de forma direta ou indireta.

Resumo

Expressões faciais são uma forma natural e expressiva de transmitir emoções e intenções humanas. Diferente do cérebro humano, o reconhecimento destas expressões não é uma tarefa trivial para um computador. Nos últimos anos, diferentes sistemas com base em redes de arquitetura profunda foram propostos para resolver problemas da visão computacional, revolucionando o estado da arte. Todavia, diferentes arquiteturas de rede apresentam diferentes características de aprendizado, mostrando-se mais adequadas para padrões específicos. Este trabalho propõe, então, uma abordagem híbrida que envolve o zoneamento da imagem de entrada, a classificação das subimagens com redes especialistas, e o reconhecimento final através de técnicas de modelagem de sequências. Como estudo de caso, aplica-se o modelo proposto ao problema de reconhecimento de expressões faciais. A partir dos experimentos, observa-se que o modelo fornece resultados competitivos com o estado-da-arte, além de oferecer alternativas para exploração da combinação de diferentes redes especialistas.

Palavras-chave: Reconhecimento de padrões, Expressões Faciais, CNN, HMM, Zoneamento.

Abstract

Facial expressions are a natural and expressive way of conveying human emotions and intentions. Unlike the human brain, recognizing those expressions is not a trivial task for a computer. In recent years, multiple deep-learning models have been proposed to solve computer vision problems, revolutionizing the state of the art. However, different network architectures have different learning abilities, making them more appropriate for specific patterns. Therefore, this study proposes a hybrid approach involving the zoning of the input image, the classification of the sub-images with expert networks, and a final recognition through sequence modeling techniques. As a case study, we apply the model to the problem of facial expression recognition. From the experiments, it is observed that the model achieves competitive results, while providing alternatives for exploring the combination of different expert networks.

Keywords: Pattern recognition, Facial Expressions, CNN, HMM, Zoning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

RNA	Rede Neural Artificial
MLP	*Multilayer Perceptron*
CNN	*Convolutional Neural Network*
HMM	*Hidden Markov Model*
FC	*Fully-Connected*
ReLU	*Rectified Linear Unit*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
DTW	*Dynamic Time Warping*
SVM	*Support Vector Machine*

Sumário

1 Introdução

1.1 Qualificação do Problema

Expressões faciais são a forma mais natural e expressiva de transmitir emoções e intenções humanas. Parte essencial da nossa interação com terceiros, essa forma de comunicação não-verbal tem atraído pesquisadores de diferentes campos de pesquisa, como nos estudos psicológicos das emoções básicas (EKMAN, 1993) e na elaboração de técnicas inteligentes de interação homem-máquina (FASEL; LUETTIN, 2003).

O cérebro humano é capaz de facilmente interpretar diferentes padrões, permitindo-nos a fácil compreensão de expressões faciais. Para um computador, todavia, essa é uma tarefa não-trivial. Diferentes técnicas de aprendizado de máquina precisam ser aplicadas, visando capacitar a identificação das mesmas. Tais sistemas de aprendizado também podem ser encontrados em diversas outras aplicações, como reconhecimento de voz, gestos, escrita e outros.

A área de visão computacional tem como objetivo a elaboração de técnicas para construir sistemas computacionais capazes de realizar processamento e compreensão de informações visuais, tais como imagens e vídeos. Análise automática de expressões faciais tem sido, por sua vez, foco de diversas pesquisas nesta área (MORISHIMA; HARASHIMA, 1993; COLMENAREZ; FREY; HUANG, 1999; CHU; ROMDHANI; CHEN, 2014).

Nos últimos anos, diferentes sistemas com base em redes de arquitetura profunda foram propostos para resolver problemas da visão computacional, revolucionando o estado da arte (GIRSHICK et al., 2014; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Do termo em inglês *Deep Learning*, redes profundas permitem que modelos computacionais, compostos por diferentes camadas de processamento, aprendam representações de dados com múltiplas camadas de abstração. Esses métodos são capazes, então, de aprender estruturas intrínsecas em grandes conjuntos de dados ao ponto de melhorarem drasticamente o estado da arte em diferentes problemas (LECUN; BENGIO; HINTON, 2015). Dentre estas redes, destaca-se a *Convolutional Neural Network* (CNN), rede especificamente projetada para o reconhecimento de padrões em imagens. Amplamente utilizada pela literatura, a mesma é aplicada em diversos problemas de reconhecimento de objetos e padrões diversos em imagens (SZEGEDY et al., 2014; VINYALS et al., 2014)

Adicionalmente, modelos estocásticos para análise de sequências, como o *Hidden Markov Model* (HMM), também vêm sendo utilizados para a classificação de expressões faciais, normalmente mapeadas como uma sequência temporal de movimentos faciais (COHEN; GARG; HUANG, 2000). Tais abordagens visam explorar a capacidade de

representação de distribuições de probabilidade sobre uma determinada sequência de observações (RABINER; JUANG, 1986), usualmente provindas da captura de diferentes quadros em um vídeo (COHEN; GARG; HUANG, 2000).

Considerando a alta capacidade de aprendizado para classificação de imagens das redes de arquitetura profunda e o alto desempenho do HMM na categorização de séries estocásticas, uma nova abordagem híbrida pode ser proposta. Através da segmentação de diferentes pontos de interesse (e.g.: boca, nariz e olhos), problemas de reconhecimento de padrão em imagens como a classificação de uma expressão facial podem ser subdivididos em problemas menores e distribuídos para diferentes redes especialistas, sendo, por fim, reagrupados como uma sequência a ser modelada e analisada pelo HMM.

1.2 Objetivos

Este trabalho tem como objetivo propor uma nova abordagem híbrida para o reconhecimento de expressões faciais através da combinação de classificadores estocásticos de cadeias com técnicas do estado-da-arte de reconhecimento de padrões em imagens digitais. Desta forma, busca-se permitir a combinação e análise das classificações de diferentes pontos de interesse da face humana de forma isolada.

1.2.1 Objetivos Específicos

- Propor a divisão de problemas de reconhecimento de padrões em imagens em subproblemas menores, focados na análise de características específicas, permitindo a aplicação de redes neurais especializadas;
- Propor a utilização de algoritmos de modelagem de séries estocásticas (e.g.: HMM) como ferramenta para classificação global dos resultados obtidos pelas subredes;
- Analisar o desempenho do modelo proposto no problema de reconhecimento de expressões faciais e avaliar sua viabilidade e benefícios.

1.3 Estrutura da Monografia

Este documento está organizado em 6 capítulos. O Capítulo 2 inicia a fundamentação teórica com a apresentação das redes neurais artificiais, com ênfase na CNN, rede de arquitetura profunda utilizada para classificação de imagens. Em seguida, o Capítulo 3 apresenta uma introdução a classificadores de sequências, estudando o HMM. O Capítulo 4 propõe, então, uma arquitetura híbrida fundamentada nas duas técnicas anteriormente citadas. No Capítulo 5, apresenta-se a metodologia, experimentos realizados e seus resultados. Por fim, o Capítulo 6 analisa as principais contribuições e trabalhos futuros.

2 Redes Neurais Artificiais

Este capítulo visa introduzir parte da fundamentação teórica deste trabalho. A Seção 2.1 apresenta uma breve introdução às redes neurais artificiais, sendo seguida pela Seção 2.2 que detalha a rede MLP. Por fim, a Seção 2.3 detalha a rede neural utilizada no desenvolvimento do projeto, a CNN.

2.1 Introdução

O cérebro humano é um poderoso sistema de processamento de informações capaz de interpretar diferentes padrões e adaptar-se ao meio ambiente, apresentando-se altamente complexo, não-linear e paralelo. Tarefas rotineiras, como o reconhecimento perceptivo visual realizado ao se reconhecer um rosto familiar, são executadas em aproximadamente 100-200ms, enquanto tarefas de complexidade muito inferior podem demorar várias ordens de grandeza a mais para serem executadas por um computador convencional (HAYKIN, 1998).

Partindo da inspiração no funcionamento das Redes Neurais Naturais, (HAYKIN, 1998) define que Redes Neurais Artificiais (RNAs) são máquinas computacionais adaptativas projetadas para modelar a maneira como o cérebro humano realiza uma tarefa em particular ou função de interesse. São um processador extremamente paralelo, distribuído, constituído de unidades de processamento simples (neurônios), que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Enumeram-se dois aspectos que as tornam semelhantes ao cérebro:

1. Um processo de aprendizagem permite a rede adquirir conhecimento a partir de seu ambiente.

2. Pesos sinápticos são utilizados para armazenar o conhecimento adquirido.

2.2 Multilayer Perceptron

Dentre as diferentes arquiteturas possíveis para RNAs, destaca-se a *Multilayer Perceptron* (MLP). Baseada em uma das primeiras RNAs a serem projetadas, o *Perceptron* de (ROSENBLATT, 1958), a rede MLP se torna um marco por passar a permitir a resolução de problemas não-linearmente separáveis, sendo capaz de aproximar qualquer função contínua (CYBENKO, 1988).

A rede MLP é formada por um conjunto de neurônios, interconectados, organizados em camadas. Cada neurônio recebe um conjunto de sinapses para entrada, caracterizados por um peso próprio, os quais são somados ponderadamente e processados por uma função de ativação restritiva, que define a amplitude do sinal de saída. Os neurônios de uma camada são conectados a todos os das camadas vizinhas (i.e.: é *fully-connected*).

Existem três tipos de camadas utilizadas em uma rede MLP. A primeira camada é a de entrada, onde seus neurônios representam as variáveis *input* do sistema. Em seguida, encontram-se uma ou mais camadas intermediárias, escondidas, que são responsáveis pela não-linearidade do sistema. (VALENÇA, 2010) sugere o uso de uma função sigmóide de ativação, como a tangente hiperbólica ou logística nesta camada. Por fim, tem-se a de saída, representando a resposta da rede com as variáveis sendo classificadas ou previstas. A Figura 1 apresenta graficamente a arquitetura descrita.

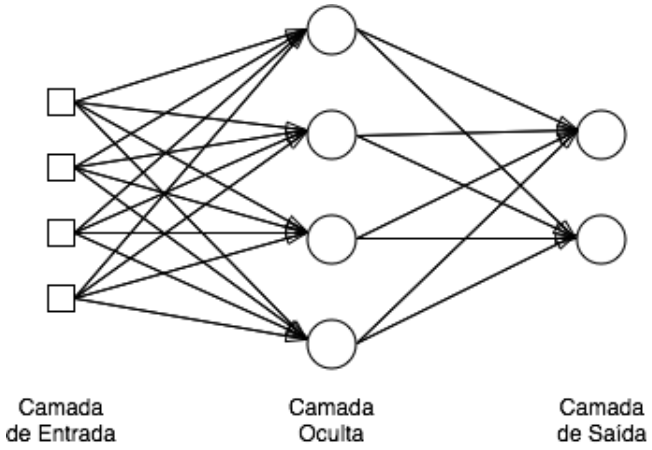

Figura 1 – Exemplo de rede MLP com quatro neurônios na camada de entrada, quatro na sua única camada oculta e dois na camada de saída.

2.3 Convolutional Neural Networks

A *Convolutional Neural Network* (CNN) funciona de forma semelhante às redes neurais artificiais tradicionais, como a MLP. Também são compostas por múltiplas camadas de neurônios, ponderados por pesos, que recebem algum sinal de entrada, calculam um produto escalar e geram uma saída através de uma função de ativação. Uma das diferenças fundamentais é que a CNN foi projetada para processar, exclusivamente, entradas multidimensionais, como imagens compostas por três vetores bidimensionais contendo a intensidade de cada pixel em cada um dos três canais de cores. Isso permite que a mesma faça suposições sobre sua entrada que a beneficiem, como: conexões locais, pesos compartilhados, *pooling* e abstração em múltiplas camadas (LECUN; BENGIO; HINTON, 2015).

Aplicações diversas de CNNs podem ser encontradas na literatura, incluindo: reconhecimento de voz (WAIBEL et al., 1989), leitura de documentos e cheques (LECUN et al., 1998), envolvendo o reconhecimento de dígitos e escrita cursiva, reconhecimento de objetos (SZEGEDY et al., 2014), reconhecimento de faces (HAMESTER; BARROS; WERMTER, 2015), descrição automática de imagens (VINYALS et al., 2014), entre outros.

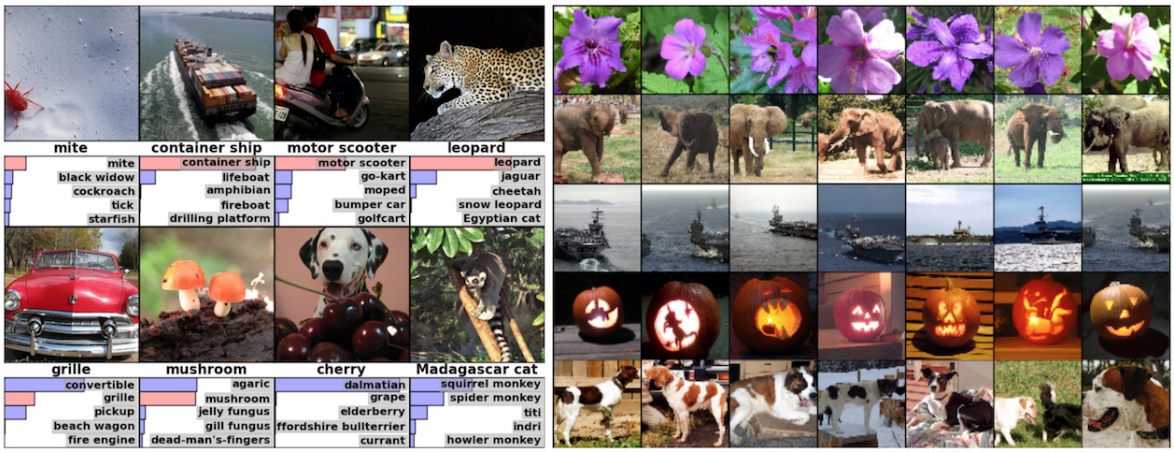

Figura 2 – Visualização do poder de classificação de uma arquitetura projetada para o *ImageNet Large Scale Visual Recognition Challenge* 2010 (ILSVRC-2010). A primeira imagem, à esquerda, apresenta oito imagens de teste e as cinco classes consideradas mais prováveis pelo modelo. A segunda imagem, à direita, apresenta cinco imagens de teste na primeira coluna, seguidas por seis colunas de imagens de treinos consideradas mais semelhantes à de teste. Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

2.3.1 Inspiração e estrutura básica

Problemas de reconhecimento de padrões em imagem notoriamente requerem que o modelo utilizado desconsidere variações irrelevantes das imagens, como mudanças de posição, orientação e iluminação, mas que, ao mesmo tempo, sejam sensíveis a detalhes específicos que de fato caracterizem o cenário presente na imagem.

Como exemplificado por (LECUN; BENGIO; HINTON, 2015) e observado na Figura 3, as fotografias de dois cachorros brancos da raça *Samoyed*, em poses diferentes, podem ser completamente diferentes se comparadas diretamente por pixel. Já a comparação das mesmas fotos com a de um Husky Siberiano branco, em pose e plano de fundo semelhantes, pode apresentar um grande número de pixels semelhantes. Como concluído por (LECUN; BENGIO; HINTON, 2015), um classificador linear ou qualquer outro classificador raso (i.e.: uma rede neural artificial de poucas camadas), operando diretamente nos pixels, não poderia distinguir as duas últimas imagens enquanto classificando as duas primeiras como da mesma classe.

Figura 3 – Três exemplos de fotografias de diferentes cachorros. As duas primeiras apresentam um cachorro da mesma raça, *Samoyed*, em diferentes poses. A última, à direita, apresenta um Husky Siberiano. Classificadores rasos, operando diretamente nos pixels, não são capazes de distinguir as duas últimas imagens enquanto classificando as duas primeiras como da mesma classe. Fonte: (DENG et al., 2009)

Percebe-se, também, problemas de escalabilidade em arquiteturas rasas, como apresentado por (BENGIO; LECUN, 2007). Para utilizar uma imagem $100 \times 100 \times 3$ diretamente como entrada de uma rede MLP, são necessários $100 * 100 * 3 = 30.000$ neurônios na camada de entrada, implicando em 30.000 pesos para conectar um único neurônio da primeira camada escondida à de entrada. Esse grande número de parâmetros acumula-se rapidamente e pode causar *overfitting*. Faz-se necessário, então, o uso de extratores de características manualmente projetados e otimizados, que busquem simplificar a imagem de entrada em valores que representem aspectos importantes para sua classificação, se mantendo invariantes para coisas irrelevantes à mesma.

Visando resolver o problema da explosão de parâmetros causada pela alta conectividade, a CNN utiliza propriedades inerentes às imagens digitais para reduzir a quantidade de pesos necessários em redes de sua profundidade. Além disso, sua modelagem busca aprender automaticamente características que identifiquem padrões na imagem, removendo a necessidade de otimização manual de extratores, que demanda extensivo conhecimento *a priori* do universo de entrada.

A arquitetura típica de uma CNN é composta por uma série de estágios, que, por sua vez, são compostos por uma ou mais camadas. Tais camadas, ao contrário da rede MLP, são organizadas em 3 dimensões: largura, altura e profundidade. O objetivo de cada uma é, então, transformar sua entrada tridimensional em uma saída também em três dimensões, através de alguma função diferenciável opcionalmente parametrizável. A Figura 4 apresenta um exemplo desta arquitetura. Ainda, nota-se que os neurônios da vasta maioria das camadas se conectam apenas a uma parte limitada da camada anterior a deles. Em geral, apenas o último estágio (chamado de *fully-connected*, ou FC) apresenta todas as conexões, por se assemelhar a uma rede tradicional acoplada para converter o aprendizado da rede em probabilidades de cada classe do problema.

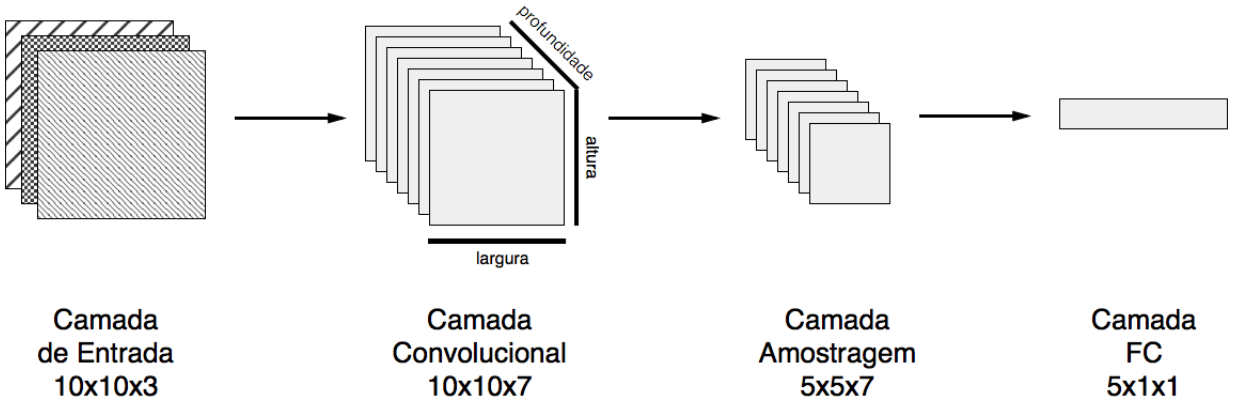

Figura 4 – Visualização de uma arquitetura CNN simples e suas camadas tridimensionais

2.3.2 Camadas na CNN

Existem quatro tipos de camadas fundamentais que estruturam os estágios de uma CNN, seguindo a camada de entrada. São elas: camada convolucional (*CONV*), função de ativação não-linear elemento a elemento (aplicada após a realização das convoluções, e.g.: *ReLU*), camada de amostragem (*POOL*) e camada totalmente conectada (*FC*). Em geral, uma arquitetura básica é composta pelo empilhamento de um estágio inicial contendo um ou mais conjuntos de camadas *CONV-ReLU-POOL*, e finalizando com a camada *FC* para cálculo dos *scores* de cada classe.

2.3.2.1 Camada Convolucional

Como visto anteriormente, CNNs exploram o princípio da conectividade local. Os neurônios da camada *CONV* se conectam apenas a uma região limitada da camada anterior. As dimensões dessa região são definidas pelo hiperparâmetro *campo receptivo*, que define a largura e altura da conectividade local. Para a profundidade, todavia, a conectividade extende-se por toda a extensão da camada anterior. Recebendo uma imagem RGB como entrada, por exemplo, um neurônio terá acesso a apenas um pedaço da imagem (largura $\times$ altura) em todos os seus canais de cores (profundidade).

A quantidade de parâmetros necessários também pode ser reduzida através do compartilhamento de pesos entre diferentes neurônios. Isso é possível partindo da suposição de que o filtro aprendido para reconhecer uma determinada característica em um pedaço da imagem também pode ser útil em outro. Por exemplo, um filtro detector de bordas do canto superior direito de uma imagem também pode detectar bordas no canto inferior esquerdo, visto que, se uma característica pode aparecer em um lugar da imagem, a mesma também pode aparecer em qualquer outro lugar (invariância de local). Pode-se denotar, então, que cada neurônio da camada convolucional compartilha seus pesos com os vizinhos no mesmo nível de profundidade, onde estes conjuntos de neurônios são chamados de *depth slices*.

Dado o compartilhamento de pesos por *depth slices*, percebe-se que o cálculo dos produtos escalares para produzir a saída da camada convolucional é, em suma, uma operação de convolução entre os pesos (aqui chamados de *filtros*, dada sua função) desta fatia pelo volume de entrada. O resultado da convolução de todos os filtros gera um conjunto de *mapas de ativação* que, ao serem empilhados em profundidade, formam o volume de saída desta camada. Isto é, a camada *CONV* processa sua entrada através de filtros, fornecendo, em sua saída, um mapeamento da ativação de cada filtro em diferentes partes da imagem. A Figura 5 apresenta um exemplo de filtros aprendidos por uma camada convolucional.

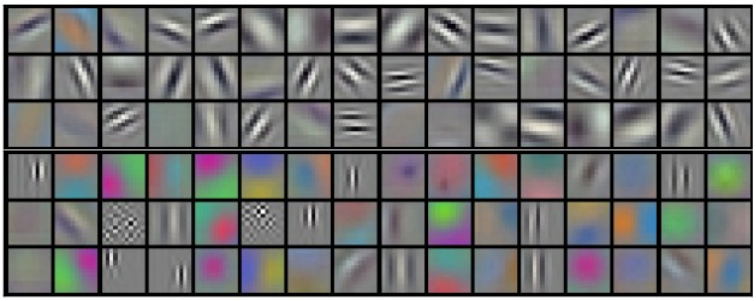

Figura 5 – Exemplo de 96 filtros convolucionais $11 \times 11 \times 3$ aprendidos pela primeira camada convolucional de uma arquitetura treinada para reconhecer objetos genéricos em imagens da ILSVRC-2010. Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Por fim, existem três outros hiperparâmetros que podem ser configurados em uma camada convolucional: *profundidade* do volume de saída, que corresponde à quantidade de filtros a serem aprendidos; *passo*, espaçamento em pixels entre o centro dos campos receptivos de cada neurônio, definindo a sobreposição dos filtros; e *zero-padding*, que pode ser utilizado para controlar o tamanho do volume de saída.

2.3.2.2 Camada de amostragem

O propósito da camada de *pooling* é fundir características semanticamente similares em uma única (LECUN; BENGIO; HINTON, 2015). A redução de tamanho (*downsampling*) das representações de entrada pode influenciar positivamente na generalização das características, ao remover pequenos deslocamentos e distorções. Além disso, a simples redução da quantidade de parâmetros ajuda a simplificar o custo computacional da rede como um todo.

A camada *POOL* é geralmente configurada através de dois hiperparâmetros: *dimensão*, que representa a largura e altura do filtro de amostragem a ser utilizado; e *passo*, que, novamente, ajusta a sobreposição da aplicação dos filtros. De acordo com (LECUN; BENGIO; HINTON, 2015), a operação de amostragem mais comumente utilizada é a

MAX, que tem como resultado o valor máximo dentre todos os valores da entrada visíveis pelo filtro.

2.3.2.3 Camada *fully-connected*

Por fim, a camada *FC* possui todos os seus neurônios conectados a todos os neurônios da camada anterior. Isto é, a mesma comporta-se como uma rede neural tradicional que recebe como entrada o resultado da ativação de cada filtro aprendido pela rede e computa a probabilidade de cada classe do problema.

2.3.3 Arquitetura Final

(LECUN; BENGIO; HINTON, 2015) afirma em seu *review* que Redes Neurais de Arquitetura Profunda, dentre elas a CNN, exploram a propriedade de que muitos sinais naturais são hierarquias de composições, nas quais características de mais alto-nível são obtidas através da composição de outras de mais baixo-nível. Um conjunto de bordas em uma imagem forma um padrão, um conjunto de padrões compõe fragmentos de objetos, um conjunto de fragmentos constrói um objeto. Na CNN, quanto maior a hierarquia de uma camada convolucional, maior o seu campo receptivo e sua capacidade de abstração.

De forma genérica, (LECUN; BENGIO; HINTON, 2015) sugere como arquitetura básica de uma CNN:

$$Input \rightarrow ((CONV \rightarrow ReLU) \times a \rightarrow POOL) \times b \rightarrow FC \times c, \tag{2.1}$$

onde a camada *POOL* é opcional e a, b e c são números inteiros não negativos que indicam a quantidade de repetições de cada estágio. Isto é: compõe-se hierarquicamente uma série de camadas convolucionais, que aprenderão filtros cada vez mais complexos e abrangentes, com camadas de amostragem que ajudam a reduzir espacialmente o volume processado. Por fim, camadas completamente conectadas computam as variáveis de saída, como o *score* das classes.

As subseções a seguir apresentam algumas arquiteturas famosas de CNNs.

2.3.3.1 LeNet

Tendo sua última iteração (LeNet-5) proposta em (LECUN et al., 1998), a LeNet é uma família de arquiteturas projetada para o reconhecimento de caracteres escritos à mão, sendo aplicada com sucesso no reconhecimento de dígitos em cheques, números de casas, entre outras aplicações. Sua arquitetura básica pode ser vista na Figura 6.

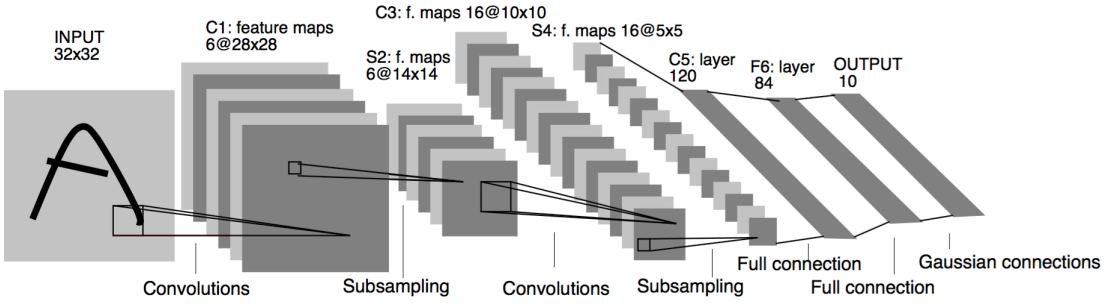

Figura 6 – Arquitetura da rede CNN LeNet-5, projetada para reconhecimento de dígitos manuscritos. Fonte: (LECUN et al., 1998).

2.3.3.2 GoogLeNet

Modelo campeão da ILSVRC-2014, competição que avalia algoritmos para detecção de objetos e classificação de imagens em larga escala, revolucionou o estado da arte ao apresentar uma nova arquitetura entitulada *Inception*. Obteve desempenho significativamente melhor que seus antecessores, usando 12 vezes menos parâmetros que a rede campeã de duas competições atrás (SZEGEDY et al., 2014). Sua arquitetura é caracterizada pela organização em módulos de *Inception* e pela sua profundidade, sendo composta por 22 camadas. A Figura 7 apresenta a estrutura da mesma.

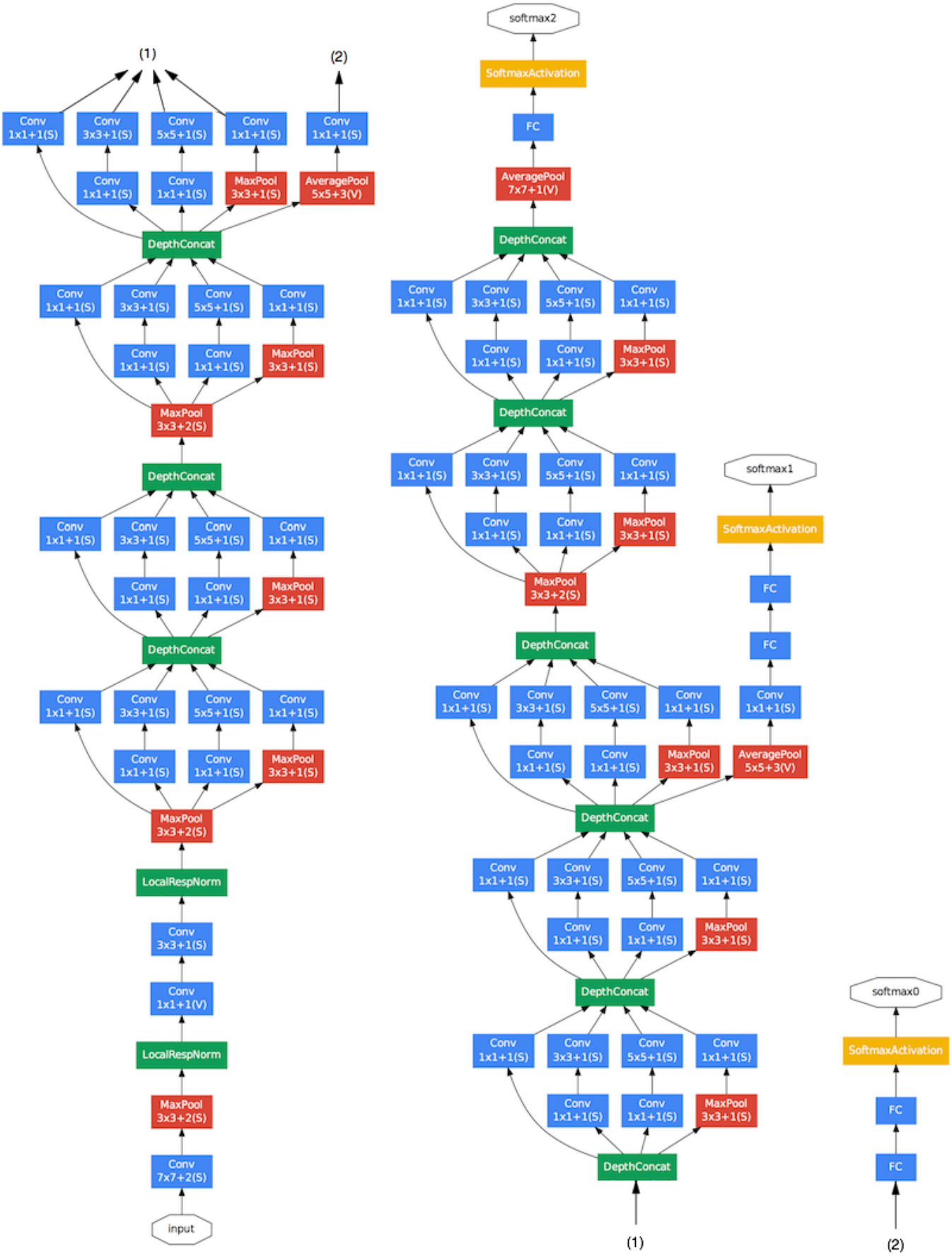

Figura 7 – Arquitetura da rede GoogLeNet, modelo campeão da ILSVRC-2014. A camada de entrada pode ser visualizada no canto inferior esquerdo. Itens (1) e (2) indicam a continuidade do fluxograma. Fonte: (SZEGEDY et al., 2014)

.

3 Classificadores de Sequências por Modelo

Classificadores de sequências possuem uma ampla gama de aplicações: classificação de sequências de proteínas do genoma, detecção de intrusão, recuperação de informação e classificação de documentos, análise de *ECG*, diferenciação entre robôs e usuários legítimos na internet, entre diversas outras áreas (XING; PEI; KEOGH, 2010). Dentre os diferentes tipos de classificadores, destacam-se os baseados em modelo, onde assume-se que a sequência foi gerada por um modelo intrínseco. Este capítulo busca detalhar o funcionamento do *Hidden Markov Model* (HMM), exemplo de classificador desta classe.

A Seção 3.1 introduz o conceito de processos estocásticos e processos Markovianos. Em seguida, a Seção 3.2 detalha o funcionamento de um HMM.

3.1 Processos Estocásticos e Modelos de Markov

Como definido por (GARDINER, 1985), processos estocásticos são sistemas em que existe uma variável aleatória dependente do tempo $X(t)$. A observação do lançamento sucessivo de moedas, os passos de uma pessoa caminhando, entre diversos outros experimentos, são exemplos de processos aleatórios.

Os Modelos de Markov são modelos que assumem a hipótese Markoviana, definida por (GARDINER, 1985) em termos de probabilidade condicional: se o tempo satisfaz a ordem

$$t_1 \geq t_2 \geq t_3 \geq ... \geq \tau_1 \geq \tau_2 \geq ..., \tag{3.1}$$

a probabilidade condicional é determinada apenas pelo conhecimento das condições mais recentes. Isto é:

$$p(x_1, t_1; x_2, t_2; ...|y_1, \tau_1; y_2, \tau_2; ...) = p(x_1, t_1; x_2, t_2; ...|y_1, \tau_1), \tag{3.2}$$

onde x_n e y_n são eventos no instante n. Nota-se que, neste exemplo, o modelo depende apenas do evento mais recente, sendo classificado como de primeira ordem.

Portanto, a probabilidade de determinados eventos acontecerem depende apenas da observação mais recente. Modelos Markovianos apresentam-se, então, como uma técnica de predição baseada apenas no atual momento de execução, já que não consideram, de forma direta, os resultados obtidos anteriormente. Como exemplo prático, (Donald Tanguay, 1995) apresenta a língua inglesa, onde a probabilidade de observar a letra "u"ao processar

uma palavra depende fortemente da letra que foi indentificada por último, visto que essa está quase sempre precedida pela letra "q".

3.2 Hidden Markov Model

(RABINER; JUANG, 1986) sugere, como motivação para o HMM, processos do mundo real que aparentam apresentar um comportamento de mudanças sequenciais: suas propriedades se mantêm aproximadamente constantes por um determinado intervalo de tempo, com pequenas flutuações, até que, em um determinado instante, há uma troca dessas propriedades. Enumeram-se, então, três problemas com esses processos:

1. Como identificar esses momentos de estabilidade e variação?
2. Como caracterizar essa natureza de evolução sequencial?
3. Qual período de tempo típico/curto deve ser escolhido para a análise?

Os HMMs tratam desse problema com sucesso por se tratarem de processos Markovianos duplamente estocásticos. Isto é, HMMs são compostos por um processo estocástico interno, não diretamente observável, que só pode ser analisado e observado através de outro processo estocástico externo que produz a sequência de símbolos visíveis.

(FOSLER-LUSSIER, 1998) exemplifica a aplicação de um HMM na previsão do tempo. Considere que existem três possíveis estados para o clima: ensolarado, chuvoso ou nublado. Para simplificação do modelo, assumiremos que o clima é estável por toda a duração de um dia. Caso possamos assumir que a previsão do tempo para um dia depende apenas da previsão do tempo do dia anterior, teremos satisfeito a hipotése Markoviana. Basta olhar o céu do dia atual para prever o clima do dia seguinte. E se não for possível observar diretamente o céu? Caso restrinjamos as observações para torná-las indiretas (e.g.: você agora está dentro de uma casa e só observa pessoas carregando guarda-chuva ou não), passamos a ter um HMM. Você não tem acesso ao real estado de seu processo, apenas a certos observáveis que podem ser emitidos: é mais provável que as pessoas carreguem um guarda-chuva em um dia chuvoso, apesar disso também poder ser feito em um dia ensolarado, nos levando a um novo conjunto de probabilidades.

Um HMM λ pode ser escrito na notação:

$$\lambda = (A, B, \pi), \tag{3.3}$$

onde:

A é a matriz de probabilidade de transições, com cada elemento a_{ij} sendo a probabilidade de sair de um estado i e ir para um estado j

π é o vetor de probabilidades de distribuição inicial, onde cada π_i representa a probabilidade do estado inicial ser i

B é a matriz de probabilidades dos observáveis, onde cada elemento b_{ij} representa a probabilidade do observável j ser emitido no estado i (NEFIAN; Hayes III, 1998). Essa matriz pode ser composta por valores discretos ou por distribuições contínuas, definindo modelos de escopo discretos e contínuos.

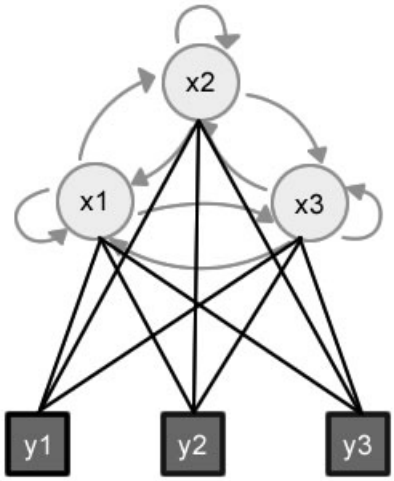

Figura 8 – Representação gráfica de um HMM, de estados x ocultos (cinza claro) e seus sinais observáveis (cinza escuro) y.

3.2.1 Problemas Canônicos

Existem três problemas canônicos para um HMM (NEFIAN; Hayes III, 1998; MITRA; ACHARYA, 2007). São eles:

Evaluation Qual a probabilidade de determinada sequência de observáveis O ter sido gerada pelo modelo λ (i.e.: $P(O|\lambda)$)? O algoritmo *Forward-backward* (BAUM et al., 1970) pode ser aplicado para obter essa informação, possuindo um custo computacional múltiplas ordens de grandeza menor que resolução por "força-bruta"(RABINER; JUANG, 1986);

Decoding Dado um modelo λ e uma sequência de observáveis O, qual a sequência de estados com maior probabilidade de ter gerado O? Esta é obtida através do algoritmo

de Viterbi (FORNEY G.D., 1973; LOU, 1995), que se apresenta próximo ao ótimo (SCHÜHLI, 2005);

Training Como devem ser treinados os parâmetros de um modelo λ de modo a maximizar $P(O|\lambda)$? Utilizando o algoritmo de treinamento Baum-Welch (RABINER; JUANG, 1986).

Através dos algoritmos relacionados aos problemas canônicos, podemos realizar a classificação e previsão de sequências de observáveis modelando-os com HMMs.

3.2.2 Implementação de um HMM Contínuo Multivariável

Partindo do conceito base discutido nas subseções anteriores, um HMM pode ser modelado para lidar com diferentes tipos de dados. As principais variações estão relacionadas a dimensionalidade e continuidade dos valores observáveis. Um valor observável pode ser monovariável (apenas uma dimensão) ou multivariável (múltiplas dimensões), além de poder ser discreto ou contínuo.

A implementação discutida nas subseções a seguir diz respeito a um HMM Contínuo Multivariável, dada sua flexibilidade quanto à entrada. Dada a natureza contínua, as matrizes de probabilidade agora incorporam distribuições contínuas de probabilidade. A Distribuição de Mistura Gaussiana é apresentada por possuir um maior poder de representação (FINK, 2007). Nesta, cada estado possui um conjunto de k distribuições normais $g_{jk}(x)$ e k pesos c_{jk}, nos quais a probabilidade de emissão é dada por $b_{jk} = \sum_k c_{jk} g_{jk}(x)$, permitindo um número maior de máximos locais em comparação com a distribuição normal individualmente, que possui apenas um máximo global.

As próximas subseções apresentam os algoritmos explanados por (FINK, 2007) e (RABINER; JUANG, 1993), matematicamente.

3.2.2.1 Forward (α)

A probabilidade de que para um determinado modelo λ, a sequência de observáveis $O_1, O_2, \ldots, O_t$ seja gerada no tempo t e o estado (s) de valor i seja alcançado, é dada pelas variáveis *forward* $\alpha_t(i)$. Isto é:

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, s_t = i|\lambda). \tag{3.4}$$

O cálculo da matriz α é feito através do seguinte algoritmo recursivo:

Inicialização Para todos os estados i:

$$\alpha_1(i) = \pi_i b_i(O_1); \tag{3.5}$$

Recursão Para todos os estados j e $t = 1 \dots T-1$:

$$\alpha_{t+1}(j) = \sum_i (\alpha_t(i) a_{ij}) b_j(O_{t+1}). \tag{3.6}$$

3.2.2.2 Backward (β)

A probabilidade de uma sequência parcial de observáveis $O_{t+1}, O_{t+2}, \dots, O_T$ a partir do tempo $t+1$, partindo de um estado j, ser gerada por um determinado modelo λ pode ser calculada através das variáveis *backward* β:

$$\beta_t(j) = P(O_{t+1}, O_{t+2}, \dots, O_T | s_t = j, \lambda) \tag{3.7}$$

O algoritmo recursivo utilizado para o cálculo da matriz β é composto por:

Inicialização Para todos os estados i:

$$\beta_T(i) = 1; \tag{3.8}$$

Recursão Para todos os estados i e $t = T-1 \dots 1$:

$$\beta_t(i) = \sum_j a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)). \tag{3.9}$$

3.2.2.3 Classificação de sequências

Partindo das matrizes *forward* e *backward* definidas anteriormente, é possível calcular a probabilidade de uma determinada sequência de observáveis O ter sido produzida pelo modelo λ (FINK, 2007). Essa operação é chamada de *Forward-Backward*:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i); \tag{3.10}$$

e:

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i). \tag{3.11}$$

3.2.2.4 Treinamento utilizando *Baum-Welch*

(FINK, 2007) apresenta *Baum-Welch* como o algoritmo mais utilizado para a otimização de HMMs. Em cada iteração, o mesmo visa encontrar um novo conjunto de parâmetros para λ onde $P(O|\lambda') \geq P(O|\lambda)$, isto é, a probabilidade de ter gerado o conjunto de observações tidas como verdadeiras para esse modelo seja maior que ou igual a probabilidade obtida pelo modelo com o conjunto de parâmetros da iteração anterior.

Os valores atualizados de cada parâmetro são calculados com base em três funções:

1. Probabilidade a posteriori para a ocorrência do estado i no tempo t.

$$\gamma_t(i) = P(S_t = i|O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \tag{3.12}$$

2. Probabilidade a posteriori da transição de um estado i para um estado j em um tempo t.

$$\gamma_t(i,j) = P(S_t = i, S_{t+1} = j|O, \lambda) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} \tag{3.13}$$

3. Probabilidade de selecionar no estado j o k-ésimo componente da distribuição mistura no tempo t para gerar a observação contínua O_t.

$$\xi_t(j,k) = P(S_t = j, M_t = k|O, \lambda) = \frac{\sum_{i=1}^{N} \alpha_t(i)a_{ij}c_{jk}g_{jk}(O_t)\beta_t(j)}{P(O|\lambda)} \tag{3.14}$$

O algoritmo, então, é executado em três etapas: inicialização, otimização e verificação. Na inicialização, define-se um modelo base $\lambda = (A, B, \pi)$ com as estimativas inicias para todos os parâmetros. As duas últimas etapas são repetidas até que o critério de parada seja atingido. O critério de parada é controlado pela etapa de verificação, onde compara-se o modelo proposto pela iteração atual com o modelo anterior: caso a probabilidade de geração das sequências tenha aumentando (i.e.: $P(O|\lambda') > P(O|\lambda)$), os parâmetros são sobrescritos e a execução continua. Caso contrário, o treinamento é encerrado. Pode-se, também, definir um limiar que determine quais diferenças são consideradas significantes.

A etapa de otimização realiza a atualização dos parâmetros do modelo λ, estimando um novo modelo $\lambda' = (A', B', \pi')$. A matriz A' é atualizada como:

$$a'_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{3.15}$$

O vetor π' é atualizado por:

$$\pi'_i = \gamma_1(i). \tag{3.16}$$

Os pesos c da distribuição mistura, assim como μ' e C' de cada distribuição normal:

$$c'_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)}, \tag{3.17}$$

$$\mu'_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k) x_t}{\sum_{t=1}^{T} \xi_t(j,k)}, \tag{3.18}$$

$$C'_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k) x_t(x_t^T)}{\sum_{t=1}^{T} \xi_t(j,k)} - \mu'_{jk}\mu'^{T}_{jk}. \tag{3.19}$$

Esse processo de otimização é executado para cada sequência de observáveis no conjunto de treino e o parâmetro final do modelo otimizado se dará pela média aritmética simples (soma do resultado em cada sequência de observáveis de todos os parâmetros correspondentes dividido pelo número de sequências no treino) (RABINER; JUANG, 1993).

4 Modelo proposto

4.1 Introdução

Sabe-se que diferentes arquiteturas de redes neurais artificiais apresentam diferentes capacidades de aprendizado e generalização em problemas diversos. Padrões a serem reconhecidos podem ser compostos por características intrínsecas que são capturadas de forma melhor por diferentes redes. (LE et al., 2011) exemplifica este efeito em seu trabalho, ao treinar diferentes neurônios para a classificação de faces humanas, corpos humanos e faces de gatos. (ERHAN et al., 2014) ao apresentar uma nova abordagem para o ILSVRC-2012, demonstra como várias arquiteturas propostas se sobressaem em diferentes classes de objetos na base de dados da competição.

A complexidade da imagem a ser analisada também possui grande impacto na capacidade de generalização da rede. As camadas mais profundas de uma CNN, por exemplo, estão diretamente ligadas a sua capacidade de criação de filtros mais complexos e abstratos (ZEILER; FERGUS, 2014). O mesmo pode ser observado em diferentes redes de arquitetura profunda, em geral ocasionando um crescimento de múltiplas ordens de grandeza na quantidade de parâmetros treináveis necessários, tornando a rede mais suscetível a *overfitting*. Por esta razão, há um constante interesse da literatura em buscar diferentes alternativas ao crescimento espacial das redes neurais (SZEGEDY et al., 2014).

Encontra-se, especialmente na literatura acerca de reconhecimento de escrita cursiva, um amplo uso de técnicas de zoneamento para classificação de padrões (Rachana R. Herekar, 2014). Nestas, a imagem a ser analisada é divida em sub-imagens intituladas zonas, onde cada uma irá conter informação local a respeito do padrão a ser analisado. Em seguida, cada zona é avaliada por um classificador de padrões, e o contexto com todas as classificações é utilizado para compor o processo de reconhecimento final.

Inspirado por essas motivações, este capítulo apresenta um modelo híbrido para o reconhecimento de padrões em imagens. Neste, o conceito de zoneamento é utilizado visando a divisão da imagem a ser analisada em sub-problemas menores, em uma abordagem *dividir para conquistar*, que podem ser classificados por redes especialistas de cada zona. Por fim, um modelador de sequências pode ser utilizado para obter-se a classificação final da imagem. A Seção 4.2 discorre a respeito da arquitetura proposta e suas etapas.

4.2 Arquitetura

O modelo proposto é composto por três estágios: zoneamento da entrada, classificação das diferentes zonas e, por último, classificação das sequências. A Figura 9 apresenta visualmente o processo executado por este modelo. As subseções a seguir detalham cada etapa, enfatizando que diferentes técnicas podem ser aplicadas em cada estágio, adaptando o modelo a problemas específicos.

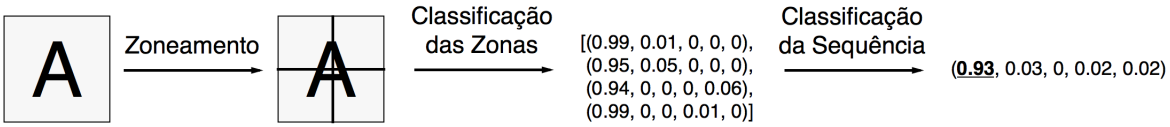

Figura 9 – Representação gráfica do processamento de uma imagem contendo a letra *A* pelo modelo proposto, visando classificá-la como uma das vogais (ou seja, 5 classes possíveis). Na primeira etapa, divide-se a imagem de entrada em quatro zonas. Em seguida, cada zona é classificada por uma rede neural produzindo tuplas de probabilidade. Por fim, o classificador de sequências produz as probabilidades finais de cada classe, mostrando que a entrada tem 93% de chance de ser a vogal *A*.

4.2.1 Zoneamento

Frequentemente utilizado em técnicas de reconhecimento de escrita cursiva, o propósito desta etapa é dividir a imagem de entrada em problemas menores, com características locais, a serem classificados por diferentes redes neurais. Desta forma, podemos reduzir a complexidade do problema e possibilitar a utilização de redes especialistas para cada zona.

(Rachana R. Herekar, 2014) define que o zoneamento pode ser classificado em topologias. As duas que se destacam são:

Topologia Estática: projetada sem o uso direto de informações previamente obtidas a respeito da distribuição das características nos padrões. Estas são usualmente propostas com base em evidências experimentais, ou experiência e intuição do projetista. Em geral, tomam forma de simples grades.

Topologia Dinâmica: obtida através do uso de técnicas de otimização, tendo como base informações específicas do problema a ser analisado.

Como resultado desta etapa, tem-se as sub-imagens obtidas através da topologia aplicada. Essas zonas possuem características locais que serão classificadas pela próxima etapa.

4.2.2 Classificação das diferentes zonas

Após a obtenção das diferentes sub-imagens que compõe a entrada, a partir da topologia escolhida, realiza-se a classificação de cada zona utilizando uma rede neural especialista, treinada para o reconhecimento das características locais da zona a qual está relacionada. Nesta etapa, as propriedades do aprendizado e poder de generalização de cada rede deve ser levado em conta. Cada zona, aqui, apresenta um novo problema, menos complexo, ao qual as arquiteturas das redes utilizadas devem se adaptar.

Nota-se que, caso a mesma rede neural seja utilizada para cada zona, a vantagem deste modelo apresenta-se apenas como redução de complexidade. Através da utilização de múltiplas redes, esta vantagem é ampliada ao permitir a utilização de redes que apresentam melhor desempenho nos problemas de cada zona. Ainda, deve-se notar que, apesar da recomendação da utilização de arquiteturas profundas, visto o alto desempenho na classificação de objetos observado na literatura, diferentes técnicas de reconhecimento de padrão também podem ser aplicados com sucesso nesta etapa.

Como resultado, temos, para cada zona, uma tupla que indica a probabilidade desta zona pertencer a uma das classes do problema. A sequência final pode ser representada como:

$$[(p_{1,1}, p_{1,2}, \ldots, p_{1,m}), (p_{2,1}, p_{2,2}, \ldots, p_{2,m}), \ldots, (p_{n,1}, p_{n,2}, \ldots, p_{n,m})], \tag{4.1}$$

onde $p_{n,m}$ indica a probabilidade da zona n pertencer a classe m.

4.2.3 Classificação das sequências

Por fim, consolida-se a série de classificações obtidas, em relação as possíveis classes do problema, através de técnicas de modelagem de sequências. Variações de *Dynamic Time Warping* (DTW), *Support Vector Machine* (SVM) e HMM podem ser utilizadas para a classificação de sequências, como analisado por (XING; PEI; KEOGH, 2010).

Nesta fase, realiza-se o treino do classificador utilizando como entrada as sequências geradas pela etapa anterior. O classificador deve ser capaz de aprender a generalizar a construção da série representante de cada classe, provendo uma maior resistência a ruído. Assim, pequenas falhas de classificação em zonas específicas podem ser corretamente ignoradas nesta etapa. Nota-se, portanto, que toda a base de dados deve ter sido convertida para este novo formato, englobando o grupo de imagens utilizadas para teste e para treino.

Como resultado, é fornecida a probabilidade da imagem de entrada pertencer a cada classe do problema.

5 Experimentos e Resultados

Este capítulo apresenta a exploração realizada acerca da aplicação do modelo proposto no Capítulo 4 ao problema de reconhecimento de expressões faciais. A Seção 5.1 detalha a motivação e realiza um breve resumo do problema a ser analisado. Em seguida, a Seção 5.2 especifica a base de imagens utilizada e as modificações realizadas à mesma. A Seção 5.3 apresenta, então, as adaptações realizadas ao modelo proposto para este problema. Por fim, a Seção 5.4 detalha os experimentos realizados, sendo seguida pelos resultados obtidos, explorados na Seção 5.5.

5.1 Reconhecimento de Expressões Faciais

Expressões faciais são a forma mais natural e expressiva de transmitir emoções e intenções humanas. Parte essencial da nossa interação com terceiros, essa forma de comunicação não-verbal tem atraído pesquisadores de diferentes campos, como nos estudos psicológicos das emoções básicas (EKMAN, 1993) e na elaboração de técnicas inteligentes de interação homem-máquina (FASEL; LUETTIN, 2003).

A positividade e negatividade emocional indiretamente transmitida por expressões faciais possue papel significativo na psicologia humana. (HAMESTER; BARROS; WERMTER, 2015) exemplifica este impacto ao analisar a teoria *broaden-and-build* proposta por (FREDRICKSON, 2001), que afirma que emoções positivas ampliam a percepção, encorajando pensamentos inovadores e exploradores, diretamente conectados a ações. Enquanto isso, emoções negativas ampliam nossa auto-consciência em relação ao ambiente.

Enfatiza-se que o reconhecimento da positividade de expressões faciais apresenta, ainda, diversas aplicações práticas de grande valor para a sociedade. Da melhoria na interação homem-máquina, como na criação de robôs inteligentes capazes de processar tais emoções, até sistemas de monitoramento de pacientes acamados, agilizando seu atendimento emergencial. A habilidade de interpretar comunicação não-verbal possui, então, grande valor para setores como saúde, automação, atendimento ao cliente, e diversos outros.

Diferentes abordagens para o reconhecimento de expressões faciais podem ser encontradas na literatura. (CHEN et al., 2011) apresenta um modelo capaz de realizar o reconhecimento a partir da combinação das informações do rosto e do corpo humano, em função do tempo. (HAMESTER; BARROS; WERMTER, 2015) propõe um novo modelo intitulado *Cross-Channel Convolutional Neural Network*, extensão da CNN para extração de características multimodais. Nota-se, todavia, que este se trata de um problema

complexo, não havendo um modelo universal que o solucione por completo.

As características intrínsecas do problema de reconhecimento de expressões faciais, como sua interpretação a partir da visualização da variação muscular em partes específicas do rosto humano, tornam este um problema propício para a aplicação do modelo proposto por este projeto. Toma-se, então, este problema como o estudo de caso deste projeto.

5.2 Base de Dados

As imagens utilizadas nos experimentos realizados por este trabalho foram extraídas do banco de expressões faciais *Cohn-Kanade* (LUCEY et al., 2010). Esta base possui 327 capturas envolvendo 123 indivíduos realizando expressões faciais mapeadas à emoções. Ao todo, 7 emoções são avaliadas: *anger*, *contempt*, *disgust*, *fear*, *happiness*, *sadness* e *surprise*. A execução de cada captura possui uma duração de até 60 quadros, onde a primeira imagem retrata neutralidade e, a última, o auge da expressão sendo realizada. Um exemplo pode ser visto na Figura 10. Para os experimentos aqui realizados, todas as imagens foram redimensionadas para 100×100 pixels.

Figura 10 – Exemplo de captura da base Cohn-Kanade. Fonte: (LUCEY et al., 2010)

.

Dado que o problema de classificação desejado relaciona-se a positividade e negatividade da expressão sendo realizada, fez-se necessária uma reorganização da base de dados, como proposta por (HAMESTER; BARROS; WERMTER, 2015). Nesta, são possíveis três diferentes classes de expressões: neutras, positivas e negativas. Em busca das imagens que melhor representam as classes, os dois quadros que compõem o auge de cada expressão são agrupados sob a classe positiva ou negativa. De forma semelhante, todas as duas imagens iniciais, onde o indivíduo não expressa emoções, classificam-se sob a neutra. As emoções ditas positivas são *happiness* e *surprised*, sendo as demais negativas.

Por fim, visando fornecer uma maior quantidade de exemplos para o treinamento dos modelos, técnicas de ampliação artificial de base de dados (*Data Augmentation*) foram

aplicadas, como sugeridas por (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Para cada imagem provida pela base, foram capturadas 9 subimagens de 96×96 pixels e 9 subimagens de 92×92 pixels, que posteriormente são redimensionadas para o mesmo tamanho da imagem original. Essa operação pode ser visualizada na Figura 11. Por fim, também calcula-se o espelhamento horizontal destas novas imagens, dobrando a quantia produzida. No total, cada imagem da base original é capaz de criar, artificialmente, 36 novos exemplos para a mesma classe.

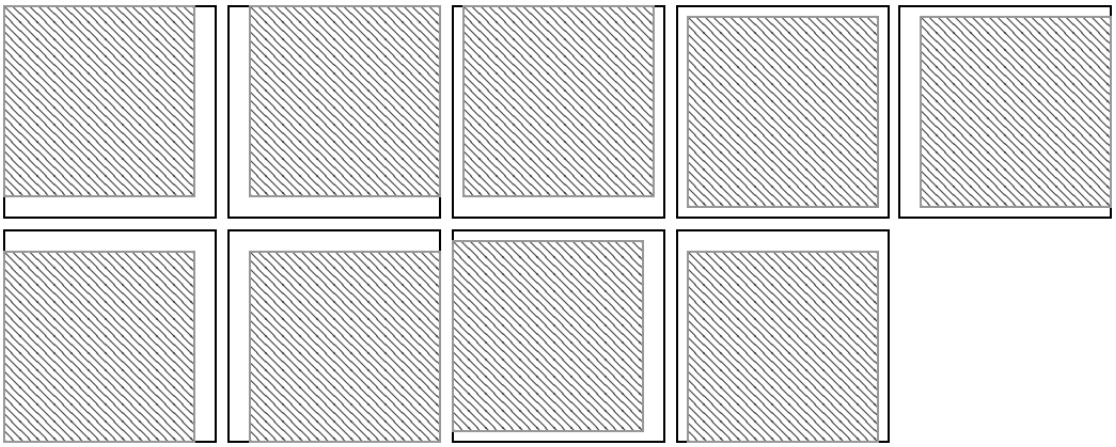

Figura 11 – Exemplo de nove posições de corte diferentes para obtenção de sub-imagens, no processo de ampliação artificial da base de dados. Os quadrados externos representam a imagem original, e os internos, com as linhas diagonais, representam a subimagem a ser obtida.

.

5.3 Adaptação do Modelo

Visando a aplicação no problema de reconhecimento de expressões faciais, realizou-se a configuração e adaptação do modelo proposto.

5.3.1 Zoneamento

A topologia de zoneamento escolhida foi a estática. Empiricamente, partindo do conhecimento da base a ser utilizada, definiu-se uma divisão da imagem em quatro faixas horizontais de 40 pixels de altura, com *overlapping* de 20 pixels. Esta separação permite que cada zona contenha o seguinte conjunto de informações, em ordem:

Zona 1 (0-40px): Testa e início das sobrancelhas, contendo informação a respeito da inclinação das mesmas;

Zona 2 (20-60px): Parte das sobrancelhas, olho, e parte superior do nariz, contendo informação a respeito da movimentação muscular de todas as regiões;

Zona 3 (40-80px): Nariz e parte superior da boca, observando todas as marcas da pele causadas pela movimentação de ambos;

Zona 4 (60-100px): Boca e queixo, permitindo a visualização da abertura da boca e movimentação do queixo.

Figura 12 – Visualização das subimagines geradas pelo zoneamento. (a) Imagem original, (b) zona 1, (c) zona 2, (d) zona 3, (e) zona 4.

.

5.3.2 Classificação das Zonas

Percebe-se um crescente interesse em diferentes arquiteturas da CNN na última década, sendo aplicadas com sucesso na detecção, segmentação e classificação de objetos e regiões em imagens (RUSSAKOVSKY et al., 2015). Seguindo os resultados promissores expostos pela literatura, a CNN foi escolhida para integralizar o sistema proposto por este trabalho em sua fase de classificação inicial.

Visto que a análise do comportamento de redes neurais diversas na classificação de diferentes zonas do rosto humano é fora do escopo deste trabalho, uma única arquitetura básica da CNN foi compartilhada por todas as zonas:

$$Input \rightarrow CONV \rightarrow ReLU \rightarrow MAX - POOL \rightarrow CONV \rightarrow ReLU \rightarrow FC, \qquad (5.1)$$

onde as camadas $CONV$ possuem 64 filtros 5×5 e 128 filtros 16×16, respectivamente, e a operação $MAX\text{-}POOL$ é realizada com configuração padrão de filtro 2×2. Por fim, a camada FC possui 12.672 neurônios de entrada e três de saída, representando as três possíveis classes: neutra, positiva e negativa.

5.3.3 Classificação das Sequências

Dentre os diferentes métodos de classificação de sequências baseados em modelos presentes na literatura, o HMM foi escolhido para esta etapa. Das características diversas que se destacam para a decisão, enfatiza-se seu poder de representação do modelo intrínseco a partir da observação de variáveis estocásticas. Através disso, pode-se modelar um

classificador robusto capaz de ponderar corretamente as variações nas probabilidades obtidas pela etapa anterior, sendo mais invariante a ruídos. Exemplos de utilização de HMMs no reconhecimento de padrões podem ser visto em (BARROS et al., 2013).

A arquitetura do HMM utilizada nos experimentos apresenta quatro estados internos, para representação intrínseca das zonas, e função de distribuição de probabilidade Gaussiana multivariável. O treinamento com *Baum-Welch* é limitado a 500 iterações.

5.4 Metodologia e Experimentos

A implementação de todo o modelo foi realizada utilizando Torch, um *framework* de computação científica com vasto suporte a algoritmos de aprendizado de máquina (COLLOBERT; KAVUKCUOGLU; FARABET, 2011). A linguagem de programação utilizada foi Lua, mantendo integração com códigos escritos para a plataforma de computação paralela CUDA (SANDERS; KANDROT, 2010). Para o HMM, em especial, optou-se por uma implementação customizada em Java, dado seu relativo baixo custo computacional, sendo integrada posteriormente à plataforma do projeto.

Visando a normalização dos dados de entrada, duas operações são realizadas no pré-processamento: padronização das características e normalização de contraste local. Neste caso, a primeira padroniza a escala dos dados de forma a terem média zero e variância unitária, enquanto a última busca reduzir o impacto causado pela diferença de iluminação e/ou cor de pele. Exemplos do pré-processamento podem ser visualizados na Figura 13.

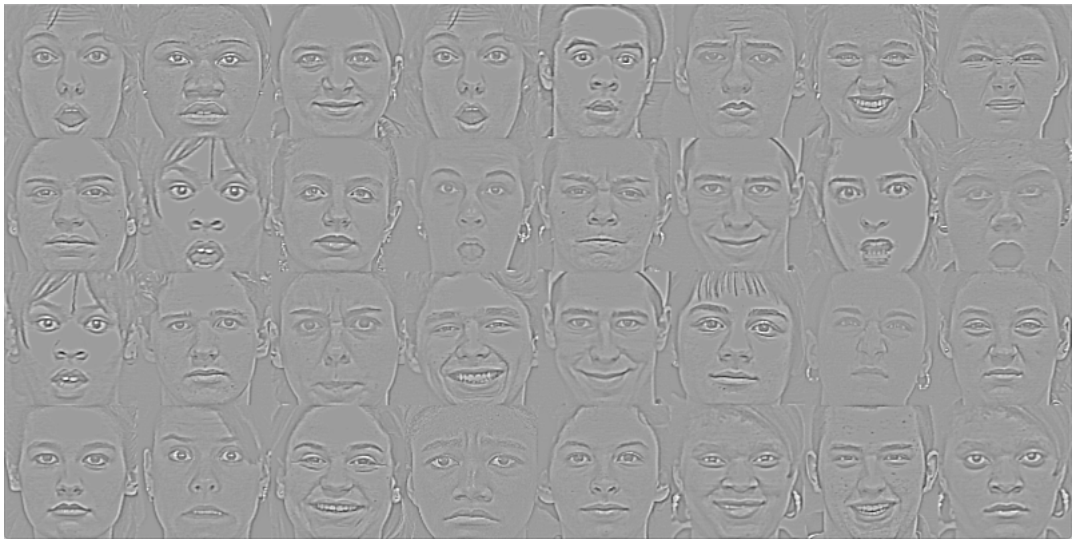

Figura 13 – Exemplo de imagens da base Cohn-Kanade após o padronização das características e normalização de contraste.

Para cada experimento realizado foram executadas 5 simulações, obtendo-se a média e desvio padrão. A seleção das imagens a serem utilizadas para treino e para teste

foram realizadas de forma aleatória. Em todos os experimentos a mesma arquitetura de CNN e HMM foram utilizadas, como apresentado nas seções anteriores. Em cada simulação, o treinamento da CNN foi limitado a 10 épocas, escolhendo-se o modelo que apresentou as melhores taxas de classificação. Três experimentos foram realizados, sendo eles:

Experimento 1 (E1): Partindo da base ampliada, 40% das imagens para treino da CNN, 40% para treino da HMM e 20% para teste individual de cada etapa e do modelo geral;

Experimento 2 (E2): Partindo da base original (i.e.: não-ampliada, limitação proposital), 40% das imagens para treino da CNN, 40% para treino da HMM e 20% para teste individual de cada etapa e do modelo geral;

Experimento 3 (E3): Partindo da base ampliada, 60% das imagens compartilhadas para treino tanto da CNN quanto da HMM (limitação proposital), utilizando o restante para teste em todas as etapas.

5.5 Resultados

A Tabela 1 apresenta a taxa de acerto média obtida para cada um dos experimentos realizados, assim como o desvio padrão. Observa-se que o experimento que obteve a maior taxa de acerto foi o E1, como esperado, visto que este não contém as limitações dos outros experimentos. O E2, por não ter realizado a ampliação da base de dados, tem a sua etapa de treinamento prejudicada, pendendo para *overfitting*.

Destaca-se, no E3, a demonstração do impacto do compartilhamento da base de treinamento entre a etapa de classificação das zonas (CNN) e a de classificação de sequências (HMM). Caso a CNN memorize a base de treino, obtendo alta porcentagem de acerto na mesma, o HMM terá poucos exemplos ruidosos para tomar como exemplo em seu aprendizado, prejudicando seu poder de generalização. Por isso, recomenda-se a separação da base de treino das duas etapas, como feito em E1.

Tabela 1 – Taxa de acerto média e desvio padrão para cada experimento realizado.

Experimento	Taxa de acerto	Desvio Padrão
E1	93,81%	1,1%
E2	73,16%	1,6%
E3	80,10%	1,2%

A Tabela 2 apresenta exemplos das taxas de acerto obtidas pela CNN (i.e.: etapa de classificação de zonas). Observa-se que o modelo proposto foi capaz de otimizar a classificação realizada pela CNN, melhorando a taxa final, nos experimentos E1 e E3.

Já no experimento E2, o modelo proposto fez com que o resultado da classificação total tendesse para a classificação da zona melhor reconhecida.

Tabela 2 – Exemplo de taxa de acerto da CNN em diferentes zonas, por experimento.

Experimento	Zona 1	Zona 2	Zona 3	Zona 4
E1	67,05%	84,04%	83,71%	91,90%
E2	62,82%	69,73%	65,13%	74,01%
E3	62,23%	65,46%	65,87%	75,71%

A Tabela 3 apresenta um exemplo de matriz de confusão obtida durante o treinamento de uma zona 1 pela CNN do experimento E3. Observa-se um grande número de erros no reconhecimento das imagens da classe Neutra. Este comportamento pode ser explicado pela falta de características determinantes para esta classe: suas imagens são, em grande maioria, genéricas.

Tabela 3 – Exemplo de matriz de confusão para a zona 1, experimento E3. As linhas representam as classes encontrada pela CNN, enquanto as colunas indicam a classificação correta. O valor indicado é referente a quantidade de exemplos da classe da coluna que foram classificados como a classe da linha.

	Negativa	Neutra	Positiva
Negativa	5658	241	2885
Neutra	928	196	2896
Positiva	932	405	7843

Por fim, conclui-se que o modelo proposto demonstra-se competitivo com o estado da arte na classificação da positividade de expressões faciais da base *Cohn-Kanade*, obtendo resultados similares aos encontrados em outros trabalhos como (BARROS; WEBER; WERMTER, 2015), como observado na Tabela 4.

Tabela 4 – Comparação entre os melhores resultados obtidos pelo modelo proposto e a CCCNN proposta por (BARROS; WEBER; WERMTER, 2015).

Modelo	Taxa de acerto	Desvio Padrão
CCCNN	92,50%	2,5%
Abordagem híbrida	93,81%	1,1%

6 Considerações Finais

6.1 Conclusões

O objetivo deste trabalho é a proposição de um modelo híbrido para reconhecimento de padrões em imagens através da combinação de classificadores estocásticos de cadeias com técnicas estado-da-arte de reconhecimento de padrões em imagens digitais, aplicando-o no problema de reconhecimento de expressões faciais. Para isto, fez-se necessário o estudo de técnicas de reconhecimento de padrão, redes neurais, redes de arquitetura profunda e classificadores de sequências.

A arquitetura aplicada nos experimentos de classificação de expressões faciais fez uso das técnicas CNN e HMM. Os resultados obtidos apresentaram-se satisfatórios, identificando classificações semelhantes ao estado-da-arte, apesar da utilização de arquiteturas básicas nas redes de suas camadas.

Deve-se salientar, entretanto, que o modelo aqui proposto não impõe limitações quanto a rede neural ou o classificador de sequências a ser utilizado. Pelo contrário, um dos seus pontos diferenciais em destaque é a possibilidade de utilização de múltiplas redes especialistas.

6.2 Trabalhos Futuros

O modelo aqui proposto possui vários pontos de extensão, além de diversas outras análises que podem ser desenvolvidas. Dentre elas:

- Avaliação do desempenho de diferentes redes neurais no problema de reconhecimento de expressões faciais, utilizando a melhor rede para cada zona;
- Availação do desempenho de diferentes classificadores de sequência, como DTW.
- Uso de zoneamento adaptativo: criação de zonas diretamente em pontos de interesse, como exclusivamente nos olhos, ou observando determinada parte da face;
- Estudo do poder de generalização através da realização de experimentos com diferentes bases de imagens.

Referências

BARROS, P.; WEBER, C.; WERMTER, S. Emotional Expression Recognition with a Cross-Channel Convolutional Neural Network for Human-Robot Interaction. *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, p. In Press, 2015.

BARROS, P. V. a. et al. Convexity local contour sequences for gesture recognition. *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, p. 34, 2013.

BAUM, L. E. et al. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, v. 41, n. 1, p. 164–171, 1970. ISSN 0003-4851.

BENGIO, Y.; LECUN, Y. Scaling Learning Algorithms towards AI To appear in Large-Scale Kernel Machines. *New York*, v. 34, n. 1, p. 1–41, 2007. ISSN 00099104.

CHEN, S. et al. Recognizing expressions from face and body gesture by temporal normalized motion and appearance features. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. [S.l.: s.n.], 2011. p. 7–12. ISSN 2160-7508.

CHU, B.; ROMDHANI, S.; CHEN, L. 3D-Aided Face Recognition Robust to Expression and Pose Variations. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, p. 1907–1914, 2014.

COHEN, I.; GARG, A.; HUANG, T. S. Emotion recognition from facial expressions using multilevel HMM. *Science And Technology*, p. 85, 2000.

COLLOBERT, R.; KAVUKCUOGLU, K.; FARABET, C. Torch7: A Matlab-like Environment for Machine Learning. In: *BigLearn, NIPS Workshop*. [S.l.: s.n.], 2011.

COLMENAREZ, A.; FREY, B.; HUANG, T. S. A probabilistic framework for embedded face and facial expression recognition. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* [S.l.: s.n.], 1999. v. 1, p. 597 Vol. 1. ISBN 1063-6919.

CYBENKO, G. *Continuous Valued Neural Networks with Two Hidden Layers are Sufficient*. [S.l.: s.n.], 1988.

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

Donald Tanguay. Hidden markov models for gesture recognition. p. 1–52, 1995.

EKMAN, P. Facial expression and emotion. *The American psychologist*, v. 48, n. 4, p. 384–392, 1993. ISSN 0003-066X.

ERHAN, D. et al. Scalable object detection using deep neural networks. *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, p. 2155–2162, 2014.

FASEL, B.; LUETTIN, J. Automatic facial expression analysis: A survey. *Pattern Recognition*, v. 36, n. 1, p. 259–275, 2003. ISSN 00313203.

FINK, G. A. *Markov Models for Pattern Recognition: From Theory to Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN 3540717668.

FORNEY G.D., J. The viterbi algorithm. *Proceedings of the IEEE*, v. 61, n. 3, p. 302–309, 1973. ISSN 0018-9219.

FOSLER-LUSSIER, E. Markov Models and Hidden Markov Models: A Brief Tutorial. *Ca Tr-98-041*, v. 1198, n. 510, p. 132–141, 1998.

FREDRICKSON, B. L. The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions. *American Psychologist*, v. 56, n. 3, p. 218–226, 2001. ISSN 0003-066X.

GARDINER, C. W. *Handbook of stochastic methods*. 1985. 101 p.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *Cvpr'14*, p. 2–9, 2014. ISSN 10636919.

HAMESTER, D.; BARROS, P.; WERMTER, S. Face expression recognition with a 2-channel Convolutional Neural Network. In: *Neural Networks (IJCNN), 2015 International Joint Conference on*. [S.l.: s.n.], 2015. p. 1–8.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, p. 1–9, 2012. ISSN 10495258.

LE, Q. V. et al. Building high-level features using large scale unsupervised learning. *International Conference in Machine Learning*, p. 38115, 2011. ISSN 10535888.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. ISSN 0028-0836.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2323, 1998. ISSN 00189219.

LOU, H.-L. Implementing the Viterbi algorithm. *IEEE Signal Processing Magazine*, v. 12, n. 5, p. 42–52, 1995. ISSN 1053-5888.

LUCEY, P. et al. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, n. July, p. 94–101, 2010. ISSN 2160-7508.

MITRA, S.; ACHARYA, T. Gesture Recognition : A Survey. *IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews*, v. 37, n. 3, p. 311–324, 2007. ISSN 10946977.

MORISHIMA, S.; HARASHIMA, H. Facial expression synthesis based on natural voice for virtual\nface-to-face communication with machine. *Proceedings of IEEE Virtual Reality Annual International Symposium*, 1993.

NEFIAN, A. V.; Hayes III, M. H. Hidden Markov models for face recognition. *Acoustics Speech and Signal Processing*, v. 5, n. 4, p. 2721–2724, 1998. ISSN 15206149.

RABINER, L.; JUANG, B. An introduction to hidden Markov models. *IEEE ASSP Magazine*, v. 3, n. January, p. 4–16, 1986. ISSN 1934-340X.

RABINER, L.; JUANG, B.-H. *Fundamentals of Speech Recognition*. 1993. 507 p.

Rachana R. Herekar. Handwritten Character Recognition Based on Zoning Using Euler Number for English Alphabets and Numerals\n. *IOSR Journal of Computer Engineering (IOSR-JCE)*, v. 16, n. 4, p. 75–88, 2014.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in .... *Psychological Review*, v. 65, n. 6, p. 386–408, 1958. ISSN 1939-1471(Electronic);0033-295X(Print).

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015.

SANDERS, J.; KANDROT, E. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0131387685, 9780131387683.

SCHÜHLI, E. S. E. Reconhecimento de gestos de maestro utilizando redes neurais artificiais parcialmente recorrentes. 2005.

SZEGEDY, C. et al. Going Deeper with Convolutions. *arXiv preprint arXiv:1409.4842*, p. 1–12, 2014. ISSN 1550-5499.

VALENÇA, M. *Fundamentos das Redes Neurais*. [S.l.]: Livro Rápido, 2010. ISBN 9788577163427.

VINYALS, O. et al. Show and Tell: A Neural Image Caption Generator. 2014.

WAIBEL, a. et al. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 37, n. 3, p. 328–339, 1989. ISSN 00963518.

XING, Z.; PEI, J.; KEOGH, E. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, v. 12, n. 1, p. 40, 2010. ISSN 19310145.

ZEILER, M.; FERGUS, R. Visualizing and understanding convolutional networks. *Computer Vision - ECCV 2014*, v. 8689, p. 818–833, 2014. ISSN 978-3-319-10589-5.