

# Hybrid Approach for Detecting Brazilian Real Coins with Localization Algorithms and Convolutional Neural Networks

David C. Yonekura, Elloá B. Guedes

<sup>1</sup>Grupo de Pesquisa em Sistemas Inteligentes  
Universidade do Estado do Amazonas (UEA)  
Av. Darcy Vargas, 1200 – Manaus – Amazonas  
{dcy.eng17, ebgcosta}@uea.edu.br

**Abstract.** *Multiple coin detection is a Computer Vision problem which aims at finding coins in an image and classifying their respective values. For Brazilian Real, literature typically consider feature extraction techniques followed by Machine Learning classifiers, specially Artificial Neural Networks. However, some recent results on coin detection with Deep Learning motivated us to propose a hybrid approach for the previously mentioned problem considering localization with traditional Computer Vision techniques and classification with Convolutional Neural Networks. Multiple architectures were tested in a realistic dataset and the proposed solution has mean average precision higher than 92% with classifier accuracy greater than 97%, surpassing other work for the same dataset by 6%.*

**Resumo.** *A detecção de múltiplas moedas é um problema de Visão Computacional que objetiva encontrar moedas em uma imagem e classificá-las quanto aos seus respectivos valores. Para o Real, a literatura tipicamente considera o uso de técnicas de extração de características seguida de classificadores de Machine Learning, em especial as redes neurais artificiais. Porém, considerando resultados recentes com Deep Learning, houve a motivação de propor uma abordagem híbrida considerando a localização de moedas com técnicas tradicionais de Visão Computacional e classificação com Redes Neurais Convolucionais. Múltiplas arquiteturas foram avaliadas e a solução proposta possui mean average precision superior a 92% com acurácia de classificação maior que 97%, superando outro resultado em 6%.*

## 1. Introdução

A detecção automática de moedas consiste em uma tarefa de Visão Computacional com vistas a localizar múltiplas moedas em uma imagem e atribuir-lhes rótulos correspondentes aos valores financeiros que representam. Tal detecção é útil para a implementação de dispositivos eletrônicos de pagamento, quer seja recebendo ou dispensando moedas, para a aceleração da contagem ou revisão de valores em transações financeiras, para a elaboração de soluções que mitiguem as dificuldades enfrentadas por aqueles que possuem limitações visuais, e até mesmo para colaborar eventualmente na identificação de moedas falsificadas ou danificadas.

Os desafios concernentes à tal tarefa residem na existência de ruído nas imagens e nas variações de rotação e de outras características visuais das moedas decorrentes do desgaste natural ou de alterações indesejadas durante o processo de fabricação. Aliado a estes aspectos, há que se mencionar o apelo financeiro, em que o mapeamento incorreto dos valores pode ensejar prejuízos. No contexto da Numismática Contemporânea, que compreende o estudo de objetos monetiformes como moedas e medalhas do Século XVII até os dias atuais, há, entretanto, algumas facilidades em comparação com outros períodos pois, graças ao processo moderno de fabricação, os padrões existentes são facilmente detectáveis por inspeção visual e

há uma grande disponibilidade de moedas, o que se mostra útil para a construção de soluções baseadas em Visão Computacional (VC) [Capece et al. 2016].

A literatura já contempla a proposição de algumas soluções para esta problemática. O uso de algoritmos extratores de características com vistas a capturar informações a respeito de cor e textura para posterior classificação residem no cerne das principais contribuições [Jain and Jain 2012, Roomi and Rajee 2015], eventualmente também compreendendo redução de dimensionalidade [Farooque et al. 2016]. Porém, resultados recentes e bastante notáveis das técnicas e modelos de *Deep Learning* têm residido na área de VC, especialmente com o uso de Redes Neurais Convolucionais (CNNs, do inglês *Convolutional Neural Networks*), o que tem proporcionado um avanço no estado da arte [Khan et al. 2018]. Em termos de aplicação na detecção de moedas, só há registros do uso da arquitetura AlexNet para o reconhecimento de moedas históricas [Kim and Pavlovic 2016] e de Euro [Capece et al. 2016].

Diante do exposto, este trabalho tem por objetivo propor uma solução híbrida para a tarefa de detecção de moedas, baseada primeiramente no uso de algoritmos tradicionais de VC para localização e seguido de uma CNN para classificação, em que serão testadas diferentes arquiteturas. Para tanto, foi considerada uma base de dados realística, com múltiplos exemplos e presença de ruído. Os resultados obtidos mostram acurácia superior a 97% na classificação e métricas de detecção da ordem de 92,73%. Na solução obtida, destaca-se o uso de uma arquitetura de CNN bastante compacta, a MobileNetV2, passível de ser embarcada em dispositivos móveis.

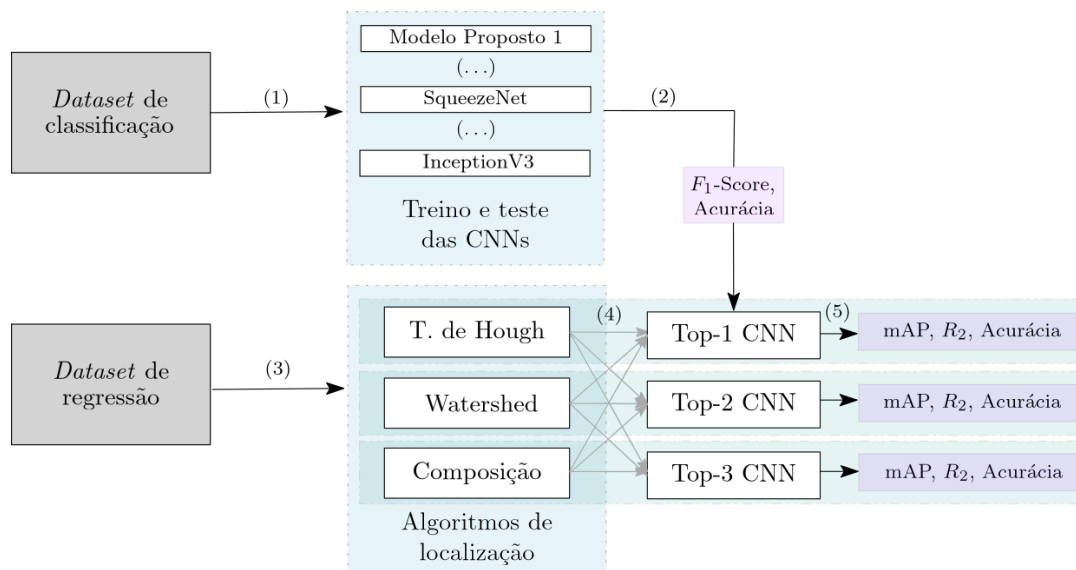
Para apresentar o que se propõe, este trabalho está organizado como segue: a metodologia do trabalho, a qual detalha a base de dados utilizada, os algoritmos, modelos e as métricas de desempenho, encontra-se apresentada na Seção 2; uma breve descrição dos trabalhos existentes na literatura voltados para moedas de Real encontram-se descritos na Seção 3; os resultados obtidos e sua discussão encontram-se dispostos na Seção 4; e, por fim, as considerações finais e sugestões de trabalhos futuros são mostrados na Seção 5.

## 2. Materiais e Métodos

A visão geral da solução proposta neste trabalho é sintetizada na Fig. 1. Primeiramente, foi considerada uma tarefa de classificação multi-classe mediante Aprendizado Supervisionado (1) para reconhecer uma moeda individualmente. Para tanto, foram treinadas e testadas diversas arquiteturas de CNNs das quais foram escolhidas as três que melhor maximizaram as métricas de desempenho (2). Em seguida, foi abordado um problema de regressão (3), o qual considerava primeiramente o uso de técnicas de localização de objetos para posterior atribuição do valor correspondente (4). A combinação do algoritmo de localização com a CNN que melhor maximizou as métricas desta tarefa (5) foi a solução de referência para o problema considerado. As subseções a seguir apresentam detalhadamente as escolhas metodológicas efetuadas para cada uma dessas etapas.

### 2.1. Dados Experimentais

Os dados experimentais considerados neste trabalho foram oriundos da base de dados *Brazilian Coin Detection Dataset* [Moneda et al. 2020], composta de 6.021 imagens coloridas nas dimensões 640px × 480px, separadas em dois conjuntos: o primeiro deles, composto de 3.056 exemplos de moedas individuais e rotuladas, com vistas à classificação (*dataset* de classificação), como ilustrado na Fig. 2a; e o segundo, composto de 2.965 exemplos com múltiplas moedas por imagem e valor total resultante como rótulo das imagens, com vistas à tarefa de regressão (*dataset* de regressão), como ilustrado na Fig. 2b. Nota-se que tais exem-



**Figura 1: Visão geral da solução proposta.**

plos não foram obtidos sob condições controladas, pois o *background* das imagens não é uniforme, há variação na incidência de iluminação, projetando sombras e, ainda, alguns exemplos possuem moedas ligeiramente sobrepostas.

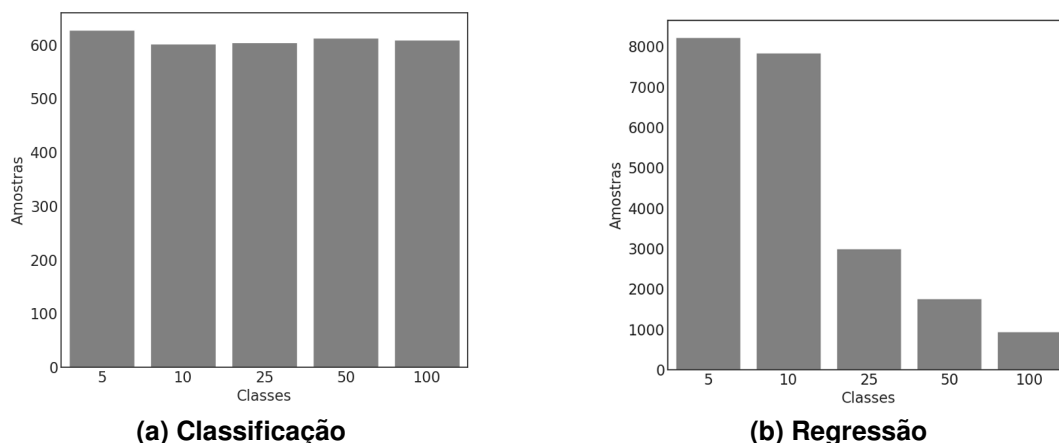


**Figura 2: Imagens da base de dados para as tarefas de aprendizado. Fonte: [Moneda et al. 2020].**

Em relação à tarefa de classificação multi-classe, o histograma da Fig. 3a ilustra a distribuição de exemplos por classe, permitindo notar que esta base de dados encontra-se balanceada. Em contraste, observa-se que não há uniformidade no quantitativo de exemplos existentes no conjunto de regressão, conforme Fig. 3b, .

## 2.2. Limpeza, Pré-Processamento e Preparação dos Dados

Primeiramente, foi efetuada uma limpeza nos dados com vistas a eliminar exemplos inadequados, nos quais as imagens não possuíam moedas ou que possuíam outros objetos alheios ao problema. Em seguida, com a porção dos dados relativos à classificação, utilizando-se das anotações produzidas, foi efetuado o recorte sob medida de cada moeda individual ali presente, preservando a referência ao rótulo associado. Como esta demarcação estava em formato circular e as imagens deveriam ser salvas em formato retangular, houve preenchimento do *background* remanescente na cor preta, seguido do redimensionamento para  $96\text{px} \times 96\text{px}$ . Esta



**Figura 3: Amostras disponíveis no *Brazilian Coin Detection Dataset* por tarefa.**

dimensão foi escolhida para tirar proveito da compatibilidade com a dimensão de entrada das principais arquiteturas de CNNs já disponíveis para utilização em *frameworks* e repositórios de modelos. No tocante à tarefa de classificação multi-classe, os exemplos resultantes foram utilizados para treinar e testar os modelos segundo uma validação cruzada do tipo *holdout*, na qual 70% dos dados foram utilizados para treino, 20% para validação e 10% para teste, segundo o paradigma de Aprendizado Supervisionado.

Considerando a porção dos exemplos presentes na base de dados relativos à tarefa de regressão, primeiramente excluiu-se todas as imagens que continham moedas individuais, pois estes exemplos eram iguais àqueles presentes no conjunto de dados de classificação, evitando assim a repetição de exemplos, o que poderia refletir em métricas de desempenho melhores mas não realísticas.

### 2.3. Algoritmos para Localização de Moedas

Para a localização de moedas, foram avaliados três algoritmos distintos, os quais são baseados em métodos clássicos de Visão Computacional:

1. **Transformada de Hough.** Técnica originalmente utilizada para encontrar retas em uma imagem, posteriormente generalizada para localizar formas arbitrárias [Ballard 1981]. Baseia-se na modificação do espaço da imagem, representando-o como uma curva no espaço de parâmetros. Em seguida, há uma votação, em que os pontos de interseção das curvas neste novo espaço são candidatos mais prováveis de possíveis objetos no espaço original [Ragab et al. 2015];
2. **Algoritmo de Watershed.** Parte de uma abordagem morfológica para segmentação e localização de objetos em imagens, combinando técnicas de crescimento de região e detecção de borda. Para tanto, agrupa os pixels em torno dos mínimos locais da imagem e os limites das regiões adjacentes seguem as chamadas linhas de crista, que dividem as zonas de influência dos mínimos [Jähne and Haußecker 2000];
3. **Composição de Operações.** Nesta terceira estratégia, já utilizada no contexto de localização de moedas pela literatura, gera-se uma imagem binária utilizando limiarização adaptativa com vistas a encontrar as bordas das moedas. Em seguida, aplica-se a operação morfológica de fechamento para assegurar a adequação das bordas e, por fim, detectam-se os contornos para posterior segmentação [Perone 2014].

Uma fração de 10% dos dados para regressão foi utilizada para efetuar um ajuste fino nos parâmetros dos três algoritmos citados, enquanto os demais dados foram reservados para

avaliação das soluções finais. Cada moeda localizada com um dos algoritmos anteriores deveria então passar por um processo de classificação do seu valor correspondente, o qual é realizado por uma CNN. Os valores obtidos das classificações serão somados e comparados ao rótulo correspondente. Porém, com vistas a maximizar o desempenho nesta tarefa, foram selecionadas as três melhores CNNs dentre as propostas na subseção a seguir.

## 2.4. Modelos, Parâmetros e Hiperparâmetros

Para a tarefa de classificação, avaliou-se diversas arquiteturas de CNNs, algumas propostas a partir da composição de diferentes camadas, e outras arquiteturas canônicas da literatura, desde aquelas com maior profundidade até aquelas voltadas para cenários com *hardware* mais restrito, tais como dispositivos móveis ou embarcados. Uma visão geral dos modelos utilizados é apresentada a seguir:

1. **Modelos Propostos 1 a 4.** Baseiam-se nos princípios elementares da composição sucessiva de camadas convolucionais e camadas de *pooling*, seguidas então de camadas completamente conectadas para classificação. Os quatro modelos possuem diferentes números de neurônios, camadas convolucionais sucessivas e tamanho dos filtros. A descrição detalhada de cada um desses modelos encontra-se na Tabela 1;
2. **MobileNetV2.** Caracteriza-se por ser uma arquitetura especialmente voltada para a utilização em dispositivos móveis. Baseia-se em uma estrutura residual invertida, em que a entrada e saída dos blocos residuais são submetidas à uma camada que restringe o número de parâmetros (*Bottleneck*), ao invés de expandi-los. Promove ainda uma redução explícita da não-linearidade nas camadas mais estreitas da rede como estratégia para melhora de acurácia [Sandler et al. 2018];
3. **SqueezeNet.** Uma arquitetura de CNN relativamente pequena composta primariamente de módulos do tipo *Fire*, os quais consistem de uma camada de compressão, a qual contém apenas filtros  $1 \times 1$ , e de uma camada de expansão, que consiste de filtros  $1 \times 1$  e  $3 \times 3$  [Iandola et al. 2016];
4. **ShuffleNet.** Arquitetura projetada para dispositivos móveis, utiliza-se de camadas do tipo *Bottleneck*, as quais reduzem a dimensionalidade da entrada, e de embaralhamento de canais nas convoluções em grupo [Zhang et al. 2018];
5. **EffNet.** Arquitetura que também baseia-se na utilização de estruturas *Bottleneck*, porém aplica convoluções que consideram a profundidade, o que as torna separáveis e paralelizáveis [Freeman et al. 2018];
6. **VGG-16.** Uma CNN canônica profunda e com grande utilização pela literatura em tarefas de Visão Computacional, compostas de blocos de 2 a 3 camadas convolucionais que utilizam filtros convolucionais  $3 \times 3$  e ativações *ReLU*, seguidas de *MaxPooling*, totalizando 16 camadas [Khan et al. 2018];
7. **InceptionV3.** É uma topologia profunda de CNN que, em contrapartida às suas antecessoras que consistiam em arquiteturas sequenciais, tornou-se o primeiro modelo que considera ramificações internas. Costuma demandar menos memória para prover resultados com alta acurácia [Khan et al. 2018].

No tocante aos hiperparâmetros, utilizou-se o *Adaptive Moment Estimation* (Adam) como otimizador do algoritmo de treinamento *backpropagation*; número máximo de épocas igual a 300; *batch size* de 64 para todas as arquiteturas, exceto para a MobileNetV2, que utilizou tal valor igual a 16; taxa de aprendizado de  $10^{-3}$  para todas as arquiteturas, exceto para a SqueezeNet e VGG16, em que este valor foi definido como  $10^{-4}$  em virtude do número de parâmetros; com o intuito de evitar *overfitting* aplicou-se a estratégia de parada antecipada

**Tabela 1: Síntese das arquiteturas dos Modelos Propostos 1 a 4, em que cada CNN encontra-se disposta verticalmente ao longo das colunas.**

Entrada: Matriz de dimensões $96 \times 96 \times 3$			
Modelo Proposto 1	Modelo Proposto 2	Modelo Proposto 3	Modelo Proposto 4
Camadas Convolucionais			
$3 \times 3 \times 8 + mp$	$3 \times 3 \times 8 + mp$	$3 \times 3 \times 16 + mp$	$3 \times 3 \times 16 + mp$
$3 \times 3 \times 8 + mp$	$3 \times 3 \times 8 + mp$	$3 \times 3 \times 32 + mp$	$3 \times 3 \times 32 + mp$
$3 \times 3 \times 16 + st + mp$	$3 \times 3 \times 16 + st + mp$	$3 \times 3 \times 32 + st + mp$	$3 \times 3 \times 64$
			$3 \times 3 \times 64$
			$3 \times 3 \times 64 + st + mp$
Camadas Completamente Conectadas			
(16,8)	(16)	(16,8)	(16,8)
Camada de Saída: 5 neurônios, função de ativação SoftMax			

mediante paciência de 40 épocas, em que a métrica utilizada no monitoramento do treino foi a acurácia da validação.

Também avaliou-se o desempenho das referidas CNNs mediante aumento artificial de dados [Chollet 2017]. Neste trabalho, tal aumento ocorreu de modo *online* por meio das seguintes transformações: rotações aleatórias de até  $180^\circ$  para a esquerda e para a direita sem ampliação e com preenchimento complementar na cor preta e variações de brilho. Estas duas transformações foram escolhidas em virtude da geometria circular das moedas e das possíveis variações na incidência de iluminação. O treino com aumento de dados utilizou um número de épocas de 400 e paciência de 80.

## 2.5. Avaliação de Desempenho

A avaliação de desempenho da solução proposta foi realizada em duas dimensões: primeiro, a avaliação individual das CNNs para tarefa de classificação e, por fim, da solução composta, que contemplava as tarefas de regressão e detecção. No tocante às CNNs individuais, a melhor classificadora foi elencada levando em conta as métricas de acurácia e  $F_1$ -Score obtidas da média de 10 execuções da validação cruzada do tipo *holdout* mencionada previamente. As repetições necessárias no método *holdout* foram adotadas com vistas a minimizar o erro estocástico decorrente da inicialização aleatória dos pesos das CNNs.

Uma vez escolhida as três melhores CNNs, houve então a aferição das tarefas de regressão e detecção. Nesta etapa, considerou-se o objetivo de identificar a composição de localizador de objetos e CNN que maximizavam as métricas para ambas as tarefas. Para a regressão, foi calculado o  $R^2$ -Score, com valor máximo igual a 1,0 [Cameron and Windmeijer 1997]. No tocante à detecção, por sua vez, a primeira métrica a ser considerada foi o *IoU* (do inglês, *Intersection over Union*), que afere a sobreposição entre a área localizada e a área de interesse, assumindo valor máximo quando esta sobreposição ultrapassa um limiar [Michelucci 2019]. Com base nos resultados possíveis em relação à este limiar tem-se a curva da Precisão *versus* Revocação. A área sob essa curva (AUC, do inglês *Area Under Curve*) para cada classe determina a métrica *Average Precision* (AP). Por último, a métrica *mAP* (do inglês, *Mean Average Precision*) sintetiza a média dos AP de todas as classes consideradas na detecção, respeitando a proporção entre os exemplos disponíveis [Everingham et al. 2010]. A acurácia também foi obtida, mas com a semântica de contabilizar o acerto de todas as moedas contidas em uma imagem. O mesmo se aplicou às métricas  $R^2$  e *mAP*, aferidas por imagem, e não por moeda

individual.

### 3. Trabalhos Relacionados

No tocante à detecção de moedas de Real, o trabalho proposto por Meneses contempla um processo que inicia com a captura das imagens, seguido da aplicação de transformada circular de Hough e de um filtro extrator de textura, para então realizar a classificação das moedas com uma rede neural artificial *feedforward multilayer perceptron* (MLP) [Meneses 2016]. Este processo permite a detecção de múltiplas moedas por imagem, é tolerante à oclusão parcial, mas depende de um aparato específico para captura das imagens. Os resultados experimentais foram conduzidos com cinco exemplares distintos de todos os tipos de moeda de Real e 400 imagens de testes contendo uma única moeda, o qual resultou em uma acurácia de 98,4%. Não foram apresentadas métricas de desempenho para cenários com múltiplas moedas e a solução é sensível à distância do dispositivo de captura em relação às moedas.

Uma outra abordagem apresentada na literatura para a detecção de moedas brasileiras contempla a utilização de uma rede neural LVQ que efetua a classificação a partir de características extraídas após a equalização da imagem de entrada, aplicação de filtro Gaussiano e localização com transformada de Hough [Mattioli and Bianco 2017]. Segundo os autores, este método é invariante à rotação e à face da moeda. A solução foi aferida perante um conjunto de teste com 100 imagens, apresentando acurácia igual a 96%.

Diferentemente das soluções anteriores baseadas em redes neurais, o trabalho de Freitas contempla a utilização de diferentes algoritmos de extração de características, tais como detecção de pontos de interesse e determinação do raio das moedas, para posterior classificação utilizando Árvores de Decisão [de Freitas 2014]. A avaliação perante um conjunto de testes contendo 364 instâncias resultou em uma acurácia igual a 90,1%. Os autores embarcaram a solução em um dispositivo móvel para auxiliar pessoas com deficiência visual.

A solução proposta no escopo deste trabalho contempla a classificação de moedas de Real utilizando CNNs que, no melhor do conhecimento dos autores desse artigo em relação à literatura, não foram ainda reportadas em outros trabalhos. Ademais, contempla uma base de dados com uma maior quantidade de exemplos e introduz também métricas de desempenho para as tarefas de localização e regressão, o que permite uma avaliação mais completa dos resultados que serão apresentados a seguir.

### 4. Resultados e Discussão

Para a condução das tarefas descritas foram elaborados *scripts* na linguagem de programação Python 3.6 com o auxílio das bibliotecas Tensorflow<sup>1</sup> e Keras<sup>2</sup>. Estes foram então executados em um servidor Intel Core i7, com 32 GB de memória principal, 960 GB de memória secundária e duas placas gráficas NVIDIA 1080 GTX com 11 GB cada, em que estas GPUs foram utilizadas para acelerar o treinamento das CNNs. As subseções a seguir descrevem os resultados obtidos.

#### 4.1. Avaliação das CNNs para Classificação Individual das Moedas

Como resultado da execução do treinamento e teste das CNNs foi possível coletar métricas de desempenho nesta tarefa, as quais encontram-se apresentadas detalhadamente na Tabela 2. Juntamente a estas, encontram-se também o número de parâmetros ajustáveis de cada modelo.

---

<sup>1</sup>Disponível em <http://tensorflow.org>.

<sup>2</sup>Disponível em <http://keras.io>.

Tabela 2: Aferição do desempenho das CNNs para tarefa de classificação.

Modelo	Padrão		Aumento de Dados		Parâmetros
	Acurácia	$F_1$ -Score	Acurácia	$F_1$ -Score	
<b>Modelo Proposto 1</b>	94,84±1,41	0,9478±0,0145	90,68±2,00	0,9061±0,0203	8.573
<b>Modelo Proposto 2</b>	95,23±1,21	0,9518±0,0125	89,65±1,19	0,8952±0,0121	8.477
<b>Modelo Proposto 3</b>	96,16±1,06	0,9614±0,0107	92,65±2,58	0,9261±0,0260	32.965
<b>Modelo Proposto 4</b>	89,45±14,52	0,8732±0,1873	93,23±2,97	0,9317±0,0301	134.501
<b>MobileNetV2</b>	97,84±0,52	0,9784±0,0052	93,26±4,92	0,9324±0,0492	2.264.389
<b>SqueezeNet</b>	96,06±1,13	0,9601±0,0114	85,00±4,66	0,8468±0,0507	725.061
<b>ShuffleNet</b>	98,19±0,77	0,9819±0,0077	91,19±3,58	0,9104±0,0385	3.533.301
<b>EffNet</b>	98,61±0,60	0,9861±0,0059	96,58±1,46	0,9658±0,0147	296.741
<b>VGG-16</b>	97,55±0,58	0,9753±0,0059	94,26±0,81	0,9420±0,0084	50.394.949
<b>InceptionV3</b>	96,06±3,90	0,9600±0,0403	90,16±5,09	0,8999±0,0529	21.813.029

Dentre as CNNs avaliadas, os Modelos Propostos 1 a 4 foram os únicos que, em algumas das repetições, apresentaram desvanecimento do gradiente descendente, produzindo previsões para uma única classe. Esses resultados não foram considerados nas métricas apresentadas por serem contraproducentes. Apesar disto, os classificadores mostraram-se bastante satisfatórios, pois na ampla maioria dos casos tanto a acurácia quanto o  $F_1$ -Score se mostraram superiores a 90%. Percebe-se que as estratégias de aumento de dados não foram eficientes em prover regularização que maximizasse o desempenho das redes. Assim, considerando o ônus computacional deste aumento, são preferíveis as soluções com desempenho compatível que se baseiam apenas nos dados originais. Ressalta-se que os melhores resultados advêm de CNNs focadas na otimização de espaço nas convoluções.

Desta etapa, destacaram-se as CNNs ShuffleNet, EffNet, VGG-16 e MobileNetV2, nesta ordem. Percebeu-se também que a VGG-16 e a MobileNetV2 possuíam desempenho equivalente e, em virtude disto, esta última foi preferida por possuir cerca de 4% do número total de parâmetros da primeira. Enfatizando a análise do desempenho dos três melhores modelos, ressaltam-se as matrizes de confusão obtidas no melhor teste, ilustradas na Fig. 4, e os gráficos Boxplot, ilustrados na Fig. 5, que denotam a variância das avaliações por modelo. Nota-se que, as medidas de tendência central e de dispersão do desempenho foram essenciais para uma avaliação de performance robusta à variação estocástica de parâmetros.

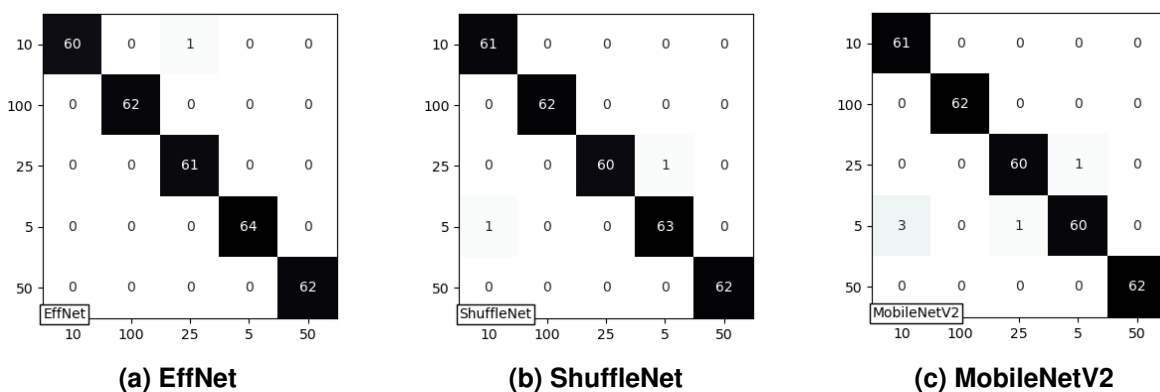
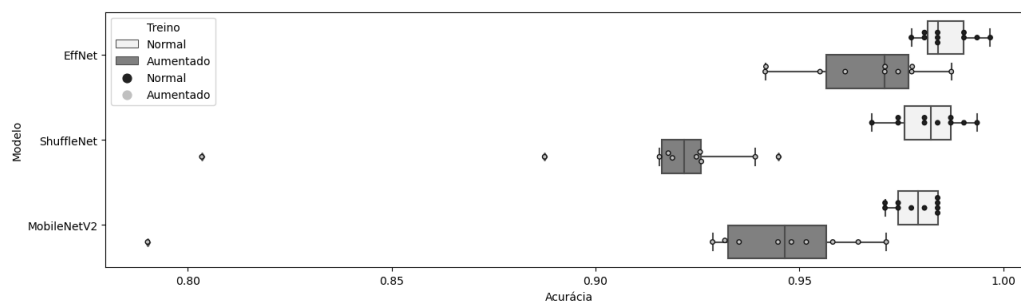


Figura 4: Matrizes de confusão das três melhores CNN avaliadas.





**Figura 5: Gráficos Boxplot da acurácia nos dez testes efetuados.**

## 4.2. Avaliação dos Algoritmos para Localização de Moedas

Considerando que os algoritmos de localização eram sensíveis à escolha de parâmetros, utilizou-se 10% dos dados de regressão para determinação dos mesmos e para uma aferição preliminar de desempenho, com resultados dispostos conforme Tabela 3.

**Tabela 3: Aferição preliminar do desempenho dos algoritmos de localização.**

Algoritmo	Acurácia (%)	<i>IoU</i>
Transformada de Hough	98,83	0,89
Algoritmo de Watershed	91,33	0,88
Composição de Operações	86,50	0,86

Em face dos resultados obtidos, foi possível verificar que o tamanho relativamente pequeno das moedas em comparação ao tamanho completo da imagem fez com que o *IoU* fosse sensível à pequenas diferenças, não ultrapassando o valor de 0,9 em nenhuma das três aferições inicialmente realizadas. A Fig. 6 ilustra algumas das particularidades de cada algoritmo na tarefa considerada. O algoritmo de Watershed, por exemplo, mostrou-se mais limitado em imagens com iluminação não uniforme. A composição de operações, por sua vez, embora tenha tido um bom desempenho em moedas individuais, falhou ao identificar como uma única moeda duas eventuais moedas que estivessem muito próximas. A transformada de Hough, por fim, obteve os resultados mais satisfatórios, com menos falhas nas situações previamente mencionadas. Uma das possíveis razões para este desempenho positivo é a transformação no espaço da imagem, permitindo uma representação mais clara dos limiares de contorno.

## 4.3. Determinação da Solução de Referência Mediante Regressão

Uma vez identificadas as melhores CNNs classificadoras e os parâmetros para os algoritmos de localização, partiu-se então para testar todas as combinações possíveis em busca da melhor solução para este cenário, a qual mostra-se híbrida por considerar técnicas clássicas da VC e o corpo de conhecimento de *Deep Learning*. Os resultados encontram-se dispostos na Tabela 4, em que a métrica de referência adotada é o *mAP*, pois a base de dados de regressão encontra-se desbalanceada e esta métrica é mais robusta à esta característica e também às diferenças dos valores das moedas. Também foram consideradas 10 repetições para minimizar flutuações estocásticas, com resultados ilustrados conforme Fig. 7.

De maneira geral, houve uma dificuldade dos modelos em diferenciar as moedas de R\$ 0,10 e R\$ 0,25 centavos, que deve ter ocorrido em razão da dificuldade de distinguir a cor de revestimento das mesmas perante as condições de iluminação não uniformes. Tal dificuldade foi refletida nos valores das métricas de *AP* e acurácia. Nota-se que o  $R^2$  é altamente

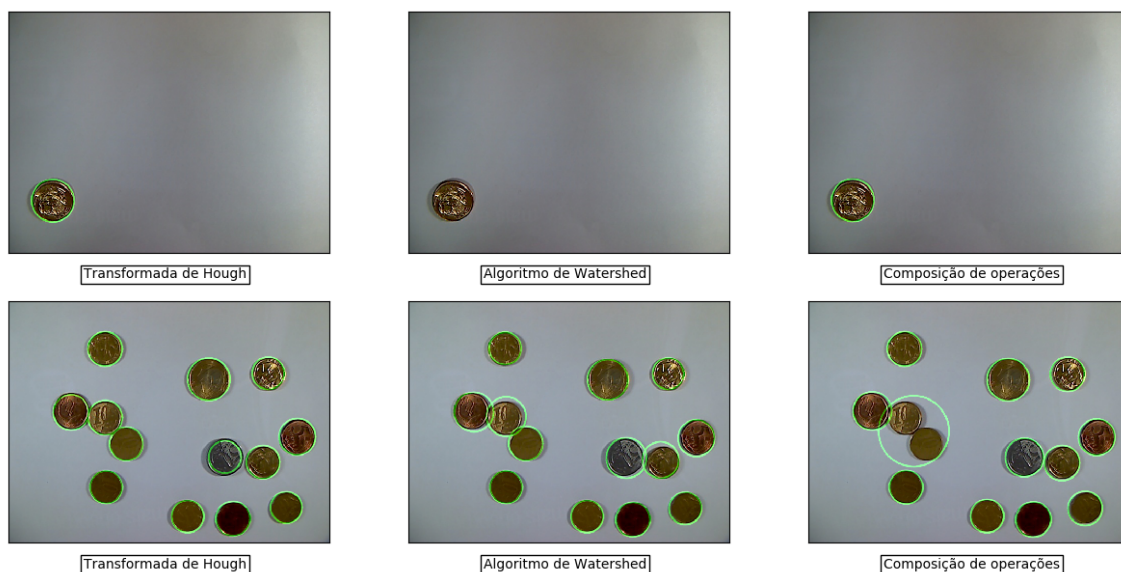


Figura 6: Cenários ilustrativos da localização efetuada pelos algoritmos considerados.

Tabela 4: Aferição do desempenho dos algoritmos de localização e CNNs.

Métricas	Hough			Watershed			Combinação		
	MobileNetV2	ShuffleNet	EffNet	MobileNetV2	ShuffleNet	EffNet	MobileNetV2	ShuffleNet	EffNet
$AP_5$	92,44	92,33	91,09	91,40	89,41	84,07	85,94	80,17	77,24
$AP_{10}$	93,47	91,78	87,45	82,33	80,85	74,42	78,66	79,52	78,19
$AP_{25}$	79,68	79,47	81,69	62,13	61,59	61,78	71,97	63,24	63,90
$AP_{50}$	98,56	96,59	97,18	98,24	93,65	96,03	91,65	77,65	86,56
$AP_{100}$	99,52	98,76	99,60	99,96	99,98	100,00	96,10	97,30	97,21
$mAP$	92,73	91,79	91,40	86,81	85,10	83,26	84,86	79,58	80,62
$R^2$	0,7440	0,6973	0,7113	0,7217	0,5620	0,6140	0,6893	0,5624	0,6430
Acurácia	0,6853	0,6631	0,5963	0,5518	0,5042	0,4108	0,5511	0,4681	0,4594

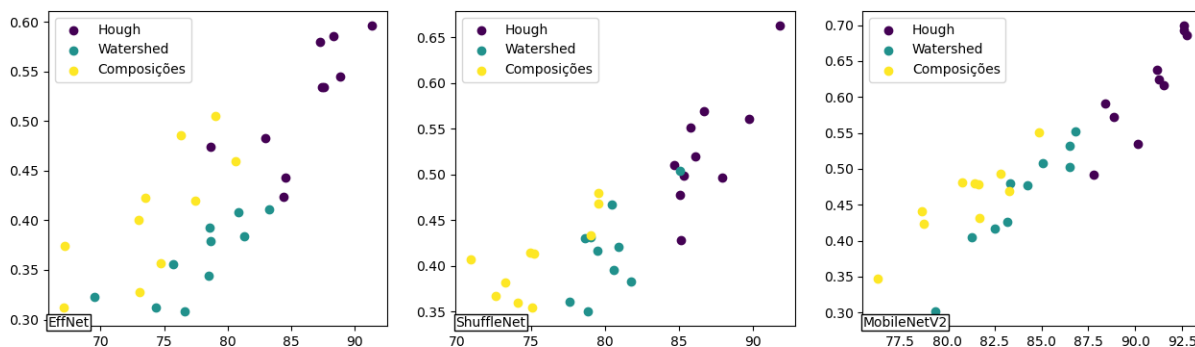


Figura 7: Scatterplot do  $mAP$  (eixo  $x$ ) versus Acurácia (eixo  $y$ ).

influenciado pela magnitude dos valores financeiros, pois um erro na detecção de uma moeda de R\$ 1,00 é penalizado 20 vezes mais que um erro na detecção de uma moeda de R\$ 0,05.

Ao final, tem-se que a solução de referência para o problema de detecção inicialmente considerado é composta pela localização com a Transformada de Hough seguida da classificação com a CNN MobileNetV2, com  $mAP$  igual a 92,73% de acerto. Se considerada apenas a classificação individual das moedas, obteve-se acurácia de 97,84%, o que mostra-se

superior a uma solução para esta mesma base de dados reportada na literatura que considera o uso de histograma de gradientes orientados e MLP, com acurácia de 92% [Gupta et al. 2019].

## 5. Considerações Finais

Este trabalho teve por objetivo propor uma solução híbrida contemplando algoritmos de localização e CNNs para detecção de moedas de Real. Utilizando-se de uma base de dados aberta, com múltiplas moedas em uma mesma imagem e presença de ruído, foram avaliadas diferentes arquiteturas de CNNs para classificação e três algoritmos para localização. A solução de referência identificada é composta pela transformada de Hough juntamente com a MobileNetV2, com  $mAP$  igual a 92,73% e classificação igual a 97,84%, em que este último resultado representa uma melhoria de aproximadamente 6% em relação ao estado da arte para a mesma base de dados.

Em trabalhos futuros pretende-se elaborar um aplicativo para disponibilizar a solução em *smartphones* e também propor e avaliar comparativamente outras técnicas de detecção de moedas, especialmente considerando o uso de R-CNNs.

## Agradecimentos

Os autores agradecem o apoio financeiro e material provido pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por meio do Programa PAIC 2019-2020 e do Projeto PPP 04/2017.

## Referências

- Ballard, D. H. (1981). Generalizing the Hough Transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.
- Cameron, A. C. and Windmeijer, F. A. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2):329–342.
- Capece, N., Erra, U., and Ciliberto, A. V. (2016). Implementation of a coin recognition system for mobile devices with deep learning. In *12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 186–192, Itália. IEEE.
- Capece, N., Erra, U., and Ciliberto, A. V. (2016). Implementation of a coin recognition system for mobile devices with deep learning. In *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 186–192, Itália. IEEE.
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications, Estados Unidos, 1 edition.
- de Freitas, S. S. (2014). Determinação do valor total de moedas em imagens digitais. Monografia (Bacharelado em Sistemas de Informação) – Universidade Federal do Ceará.
- Everingham, M., Gool, L. V., Williams, C. K., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338.
- Farooque, G., Sargano, A. B., Shafi, I., and Ali, W. (2016). Coin recognition with reduced feature set sift algorithm using neural network. In *2016 International Conference on Frontiers of Information Technology (FIT)*, pages 93–98, Paquistão. IEEE.
- Freeman, I., Roese-Koerner, L., and Kummert, A. (2018). Effnet: An efficient structure for convolutional neural networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 6–10. IEEE.

- Gupta, S., Kaur, G., Gupta, D., and Jindal, U. (2019). Brazilian coins recognition using histogram of oriented gradients features. *Journal of Computational and Theoretical Nanoscience*, 16(10):4170–4178.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50× fewer parameters and < 0.5 mb model size. Disponível em <https://arxiv.org/abs/1602.07360>. Acesso em 24 de agosto de 2020.
- Jain, N. and Jain, N. (2012). Coin recognition using circular Hough transform. *International Journal of Electronics Communication and Computer Technology*, 2(3):101–104.
- Jähne, B. and Haußecker, H. (2000). *Computer Vision and Applications – A Guide for Students and Practitioners*. Academic Press, Estados Unidos, 1 edition.
- Khan, S., Rahmani, H., Shah, S. A. A., and Bennamoun, M. (2018). *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan & Claypool Publishers, Estados Unidos.
- Kim, J. and Pavlovic, V. (2016). Discovering characteristic landmarks on ancient coins using convolutional networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1595–1600, México. IEEE.
- Mattioli, L. R. and Bianco, A. F. (2017). Reconhecimento de moedas brasileiras usando rede LVQ. In *Proceeding Series of the Brazilian Society of Applied and Computational Mathematics*, volume 5, pages 010497–1–010497–2, Gramado, RS. SBMAC.
- Meneses, G. M. (2016). Contador automático de moedas utilizando redes neurais artificiais. Monografia (Bacharelado em Engenharia da Computação) – Universidade de Brasília.
- Michelucci, U. (2019). *Advanced Applied Deep Learning – Convolutional Neural Networks and Object Detection*. Apress, Estados Unidos.
- Moneda, L., Yonekura, D. C., and Guedes, E. B. (2020). Brazilian coin detection dataset. Disponível em <http://dx.doi.org/10.21227/kj4c-y809>. Acesso em 24 de agosto de 2020.
- Perone, C. S. (2014). Simple and effective coin segmentation using Python and OpenCV. Disponível em <http://blog.christianperone.com/2014/06/simple-and-effective-coin-segmentation-using-python-and-opencv/>. Acesso em 24 de agosto de 2020.
- Ragab, M. E., Hassanein, A. S., Mohammad, S., and Sameer, M. (2015). A survey on Hough Transform, theory, techniques and applications. *International Journal of Computer Science Issues*, 12(1):139–156.
- Roomi, S. M. M. and Rajee, R. B. J. (2015). Coin detection and recognition using neural networks. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–6, Índia. IEEE.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, Salt Lake City, Utah, Estados Unidos. IEEE Press.
- Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, Salt Lake City, Estados Unidos. IEEE.