MINERAÇÃO DE DADOS COMPLEXOS

Curso de Extensão

UNICAMP

INF-0619—Projeto Final

Professor Zanoni Dias

Identificação de Empresas

Dezembro/2021

Grupo G14

Amir Carlessi

Erika Midori Kato

Fábio Seitoku Nagamine

Marcelo Cesar Cirelo

Marcos Flavio Hieda

1 Introdução

O objetivo do trabalho é construir um classificador capaz de determinar se em um endereço encontra-se uma empresa ou não. Para tanto, são fornecidos um endereço e a respectiva imagem obtida do Google Street View.

Existem inúmeras finalidades para essa tarefa de classificação, dentre elas a possibilidade de se identificar automaticamente possíveis erros em informações cadastrais de empresas que podem levar a investigações de fraudes.

2 Análise dos Dados

A base de dados fornecida contém dois arquivos de texto (‘dados_empresas.csv’ e ‘dados_nao_empresas.csv’) e 11.004 imagens (separadas em duas pastas: empresas e não empresas).

2.1 Base de Imagens

A base de imagens é composta por 11.004 arquivos, sendo 4.547 (41%) pertencentes à classe empresa e 6.457 (59%) à classe não empresa. Os arquivos correspondem a imagens extraídas do Google Street View e possuem resolução de 640x500 pixel com três canais de cor.

As imagens foram redimensionadas para 224x224 pixels, três canais, para o treino e testes dos classificadores. Essa dimensão foi escolhida para compatibilidade com a camada de entrada das redes pré-treinadas com a base de dados ImageNet.

Tabela 1. Exemplos de imagens com suas respectivas classes

Empresa	Não Empresa	Empresa	Não Empresa
			

2.2 Base de Endereços

Existem dois arquivos com informações de endereço, um para cada classe. Os atributos são diferentes nos arquivos.

O arquivo ‘dados_empresas.csv’ possui 8.000 registros, sendo que cada registro possui os seguintes atributos: ID, Nome Fantasia, Tipo Logradouro, Nome Logradouro, CEP, Município, Regime de Apuração e Endereço Completo.

Todos os registros possuem o mesmo valor para o campo “Regime de Apuração”.

O arquivo ‘dados_nao_empresas.csv’ possui 8.728 registros, com os seguintes atributos: ID, Endereço Completo. Note que este arquivo não possui a informação de CEP, necessária para a tarefa de classificação. Por isso, esse campo precisou ser derivado a partir do endereço.

Como a numeração que forma o CEP é organizada de maneira estruturada, de forma que cada dígito possui um significado específico, optamos por desmembrar o campo CEP em seus 6 componentes: região (1° dígito), sub-região (2° dígito), setor (3° dígito), subsetor (4° dígito), divisão de subsetor (5° dígito) e sufixo de distribuição (6° a 8° dígitos).

Como havia mais registros na base de endereços que na base de imagens, os registros adicionais (sem imagens equivalentes) foram removidos.

2.3 Separação em Bases de Treino, Validação e Testes

A base de dados originalmente disponibilizada não possuía separação em treino, validação e testes, sendo necessário, em acordo com o outro grupo que trabalhou com a mesma base, propor uma separação. Foi definida uma listagem de IDs para compor a base de testes. A proporção final das bases é apresentada na Tabela 2.

Tabela 2. Distribuição dos registros em bases de treino, validação e testes

Base	Empresas	Não Empresas	Proporção
Treino	2.728	3.874	60%
Validação	909	1.291	20%
Testes	910	1.292	20%
Total	4.547	6.457	100%

2.4 Aumentação da Base de Imagens

Nas tarefas de aprendizado de classificadores em bases de imagem é comum utilizar a técnica de aumento de dados como forma de tornar o classificador mais robusto. São utilizadas bibliotecas de manipulação de imagens para gerar imagens artificialmente a partir de imagens da base de dados de treino.

No nosso caso, é sabido que as imagens são provenientes do Google Street View a partir dos endereços fornecidos. Portanto, consideramos que uma abordagem mais interessante seria utilizar a própria API da Google para baixar imagens do mesmo endereço, porém em ângulos diferentes.

Para cada imagem original, foram extraídas, via API, mais três:

- Ângulo de visão (field of view, FOV) de 50°;
- Ângulo de visão (field of view, FOV) de 120°;
- Inclinação vertical (pitch) de 14°.

Tabela 3. Exemplo de ângulos diferentes para a mesma fachada.

Original	FOV 50°	FOV 120°	Pitch 14°
			
			

Nota: *em alguns casos houve alteração na fachada entre as datas de extração das imagens da base original e da base aumentada, como evidenciado na segunda linha (nas fotos mais recentes, o muro está marrom).*

2.5 Considerações sobre a Qualidade dos Dados

A base de dados original possui algumas características que necessitaram de atenção especial dos analistas. Em especial verificamos:

- Erros de anotação das imagens;
- Problemas na orientação da câmera: imagem mostra parte da rua ou do céu ao invés de uma fachada;
- Oclusão da fachada por veículos ou vegetação;
- Imagens em branco;
- Endereços sem numeração;
- Estabelecimentos/residências diferentes no mesmo endereço e, portanto, com mesma imagem (geralmente prédios e condomínios de uso misto comercial-residencial).

Apesar desses registros serem potencialmente "problemáticos" decidimos mantê-los na base de treino. Isso porque são situações reais com as quais o classificador deverá estar preparado para lidar. Além disso, a quantidade desses registros não é significativa, ficando abaixo dos 3% da base total.

3 Construção dos Classificadores

Nesta seção apresentamos o processo para identificação do melhor classificador. Buscamos utilizar todas as informações disponibilizadas pelas bases de imagens e pelas bases de endereços.

3.1 Classificadores Utilizando a Base de Imagens

Utilizamos a Rede Neural Convolucional (CNN, em inglês) nos classificadores treinando com as imagens de fachadas.

3.1.1 Utilização de CNN simples com todos os parâmetros treináveis e inicialização aleatória

O candidato inicial para classificador *baseline* foi uma rede CNN simples, de apenas três camadas, cujo treinamento utilizou apenas as imagens originais.

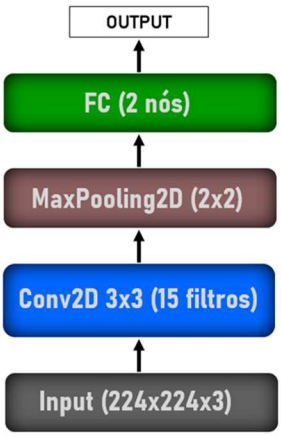


Figura 1. Arquitetura da rede CNN utilizada no classificador *baseline* inicial.

As imagens passaram por um simples pré-processamento para normalizar o valor de intensidade dos pixels. A camada de saída é uma *Fully Connected* (FC) com dois neurônios. As duas classes de saída foram codificadas usando *one-hot encoding*.

Tabela 4. Resultados de desempenho do classificador *baseline* inicial obtidos sobre a base de validação.

Modelo	Acurácia Balanceada	AUC-ROC
CNN Simples	0.69	0.75

Com o evoluir do trabalho ficou claro que a abordagem de treinar a partir de parâmetros inicializados aleatoriamente uma rede CNN, mesmo com uma estrutura simples, não seria viável. Consideramos que o desempenho era

muito baixo até para ser considerado nosso *baseline*. Por essa razão, voltamos os esforços para técnicas de transferência de aprendizado.

3.1.2 Transferência de Aprendizado

Treinar uma rede CNN com muitas camadas exige grande esforço computacional, além de bases de treino excepcionalmente grandes. Para contornar essas exigências, é comum a utilização de redes CNN pré-treinadas.

Utilizamos redes pré-treinadas com a base de imagens ImageNet que estão disponíveis no pacote Keras. O site https://keras.io/api/applications/ lista atualmente 26 modelos de redes, comparando sua acurácia com relação à base ImageNet, além de número de parâmetros e tempo de execução em uma máquina de referência. As cinco redes que escolhemos para este trabalho foram: Xception[1], VGG[2] 16, VGG-19, ResNet-50, ResNet-101.

Para adaptar as redes à nossa base de dados testamos as três abordagens mais comuns de transferência de aprendizado: utilização da rede pré-treinada com parâmetros congelados como extrator de Atributos, utilização de rede pré-treinada congelada ligada à camada de saída treinável e *fine tuning* dos parâmetros de rede pré-treinada.

As imagens foram pré-processadas com a função própria para cada rede, disponibilizada pelo Keras.

3.1.2.1 Modelo Pré-Treinado Congelado como Extrator de Atributos

Utilizamos a rede ResNet[3] 50 com todas as suas camadas bloqueadas para treinamento. Na sequência, foi adicionada uma camada GlobalAveragePooling2D (treinável). Os parâmetros dessa camada serviram de input para um classificador baseado em SVM com Kernel Gaussiano.

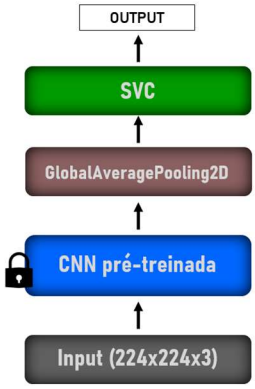


Figura 2. Arquitetura da rede CNN pré-treinada ResNet-50 sendo utilizada com entrada para um classificador SVM.

Nos testes realizados, a acurácia balanceada ficou pouco acima de 0.51, por isso consideramos que esse caminho não seria muito promissor.

3.1.2.2 Modelo Pré-Treinado Congelado Ligado à Camada de Saída Treinável

De forma semelhante à arquitetura anterior, utilizamos também redes pré-treinadas com a base de dados ImageNet com todas as camadas bloqueadas para treinamento. Na saída da rede, foi adicionada uma camada GlobalAveragePooling2D (treinável) e, por fim, a camada de saída *Fully Connected* (treinável).

---

[1] Chollet, François. "Xception: Deep learning with depth wise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
[2] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
[3] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

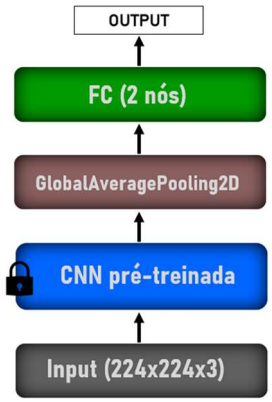


Figura 3. Arquitetura da rede CNN pré-treinada ResNet-50, ResNet-101, VGG-16, VGG-19 e Xception com camadas as bloqueadas para treinamento.

Tabela 5. Resultados de desempenho do classificador obtidos sobre a base de validação. Treino de 20 épocas.

Rede	Acurácia Balanceada	AUC-ROC
ResNet-50 (*baseline*) ☆	0.82	0.90
ResNet-101	0.80	0.89
VGG-16	0.79	0.87
VGG-19	0.77	0.86
Xception	0.71	0.81

Nota: *nesta tabela e nas próximas o ícone da estrela indica o melhor resultado considerando a acurácia balanceada. Nos casos de empate, foi escolhido o resultado com maior AUC-ROC.*

Definimos como novo *baseline* o modelo que obteve o melhor desempenho dentre os listados acima.

3.1.2.3 *Fine Tuning* dos Parâmetros de Modelo Pré-Treinado

Mudamos as configurações apresentadas no item anterior de forma a permitir o *fine tuning* dos parâmetros utilizando a base de treinamento original (TR_ORIG) e a base de treinamento aumentada com as imagens extras extraídas do Google Street View (TR_ORIG + EXTRAS).

O otimizador utilizado foi o SGD com parâmetros *learning rate* = 0.0001, *momentum* = 5 e *nesterov* = Verdadeiro. A função de *callback* utilizada foi a *EarlyStop* com paciência igual a 10, sendo que a métrica monitorada foi a acurácia calculada sobre a base de validação (*val_accuracy*).

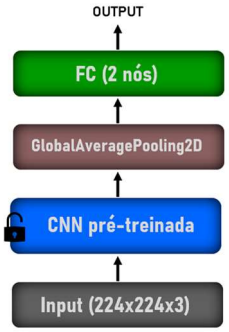


Figura 4. Arquitetura da rede CNN pré-treinada ResNet-50, ResNet-101, VGG-16, VGG-19 e Xception com parâmetros treináveis.

Tabela 6. Resultados de desempenho do classificador obtidos sobre a base de validação. Treino de 25 épocas.

Rede	Treino Original (TR_ORIG)		Treino Aumentado (TR_ORIG + EXTRAS)	
	Acurácia Balanceada	**AUC-ROC**	**Acurácia Balanceada**	**AUC-ROC**
ResNet-50	0.81	0,90	0.84	0.92
ResNet-101	0.82	0.91	---	---
VGG-16	0.84	0.92	0.85	0.93
VGG-19 ☆	0.84	0.92	0.86	0.93
Xception	0.66	0.79	---	---

Notas: *Devido ao maior tempo de treinamento demandado, apenas as 3 redes com melhores resultados (ResNet-50, VGG-16 e VGG-19) foram treinadas com a base aumentada.*

3.1.3 Melhorias nas Configurações da Rede VGG-19

A partir dos resultados obtidos no item anterior, escolhemos a VGG-19 para explorar as possibilidades de aumentar a acurácia utilizando balanceamento, mudança do otimizador, regularização L2 e variando o *learning rate*. O método de transferência de aprendizado utilizado em todos os testes foi o *fine tuning*.

Tabela 7. Variações nas configurações de treinamento da rede VGG-19

	Balancea-mento	**Otimizador**	**Regular. l2**	**Lr**	**Treino Aumentado**	**Acurácia Balanceada**	**AUC**
1	Não	Adam	Não	0.001	Não	0.82	0.82
2	Não	Adam	Não	0.0001	Não	0.85	0.92
3	Não	SGD	Não	0.00005	Não	0.83	0.92
4	Não	Adam	Não	0.0001	Sim	0.87	0.94
5 ☆	Sim	Adam	Não	0.0001	Sim	0.88	0.94
6	Sim	Adam	Sim	0.0001	Sim	0.87	0.94
7	Sim	Adam	Sim	0.0001	Não	0.85	0.93

Notas: *o balanceamento buscou pesos que buscassem dar a mesma importância a cada uma das classes; o otimizador foi executado com parâmetros β_1=0.9 e β_2=0.999.*

A alteração das configurações teve pouco impacto. A exceção foi para o *learning rate*, que quando aumentado em uma ordem de grandeza reduziu significativamente a acurácia.

3.2 Classificadores Utilizando a Base de Endereços

Para classificação utilizando a base de endereços, o CEP foi desmembrado em seus componentes, gerando os atributos: “regiao”, “subregiao”, “setor”, “subsetor”, “divisorsubsetor”, “sufixodist”.

A partir dessa base, foi treinada uma *Random Forest* com 100 árvores e *cross validation* igual a 10.

Tabela 8. Resultados de desempenho do classificador obtidos sobre a base de validação.

Classificador	**Acurácia Balanceada**	**AUC-ROC**
Random Forest (n=100)	0.88	0.88

Adicionalmente, testamos utilizar os campos Tipo Logradouro e o número (que faz parte do Endereço Completo). Porém, não houve alteração significativa no resultado.

3.3 Ensemble

Como tínhamos à disposição várias redes CNN para a classificação de imagens, além da *Random Forest* para a classificação da base de endereços, consideramos a utilização de *ensemble*.

Inicialmente, foi tentado um *ensemble* com um classificador SVM cuja entrada é a predição de 5 classificadores de imagem (baseados em CNN pré-treinada ResNet-50, ResNet-101, VGG-16, VGG-19 e Xception) e um classificador de endereços (*Random Forest*). Mas o resultado desse *ensemble* ficou inferior ao resultado dos melhores classificadores individualmente.

Tabela 9. Resultados de desempenho do classificador obtidos sobre a base de validação.

Classificador	**Acurácia Balanceada**	**AUC-ROC**
Ensemble (SVM)	0.83	0.83

Em seguida, foi realizada uma análise de quais classificadores apresentavam maior divergência entre si. O objetivo desse critério foi aumentar a variabilidade de cada componente do *ensemble*. A figura 5 apresenta os resultados das comparações de resultados sobre o conjunto de validação entre pares de classificadores.

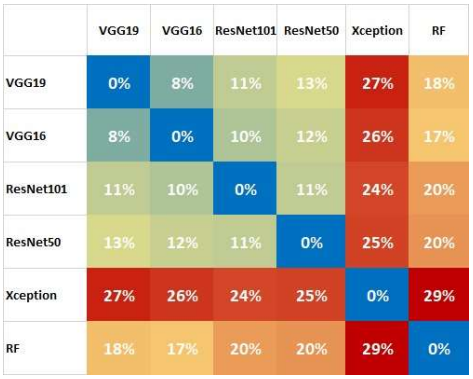


Figura 5. Comparação entre pares de classificadores.

A partir da seleção dos 3 classificadores com maior divergência (Xception, *Random Forest* e VGG-19), foram testados diversos agregadores:

3.3.1 Votação

Foram implementados três métodos de votação:

- *Maioria simples*: é escolhida a classe retornada por ao menos dois classificadores.
- *Por consenso*: a classe "não empresa" é escolhida apenas se os três classificadores retornarem "não empresa".
- *Voto de pelo menos um (minoria)*: classe "não empresa" é escolhida se qualquer classificador retornar "não empresa".

3.3.2 Soma de ativações

É escolhida a classe que tiver o maior valor da soma das ativações dos três classificadores.

3.3.3 Agregador SVM

Um classificador SVM foi treinado a partir das ativações dos três classificadores que compõem o *ensemble*.

3.3.4 Agregador MLP

De forma similar ao item anterior, um classificador baseado numa rede neural MLP de três camadas foi treinado a partir das ativações dos três classificadores que compõem o *ensemble*.

Foram avaliadas diversas variações de rede MLP (alterando a quantidade de nós, camadas e configurações de regularização), sendo escolhido o modelo que alcançou melhor resultado de acurácia balanceada sobre o conjunto de

validação. Alguns exemplos de variações de MLP avaliadas foram mantidos nas seções 4.2.6 a 4.2.8 do *Notebook* entregue junto a este relatório.

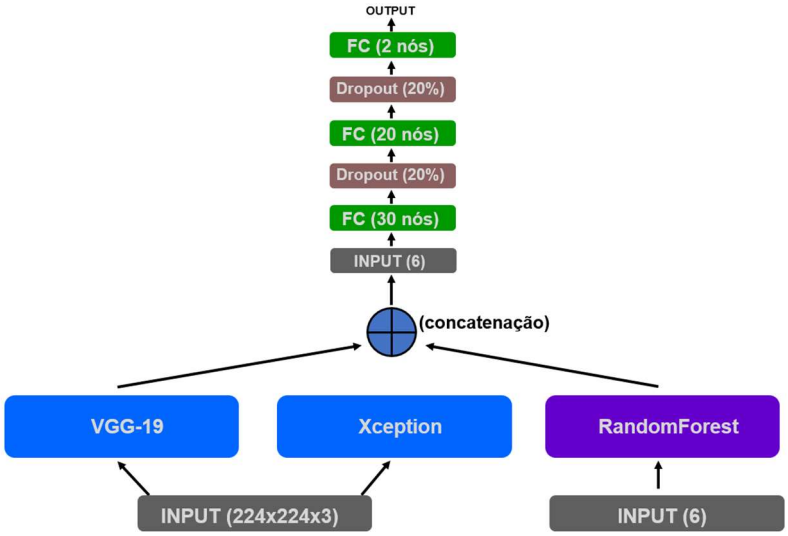


Figura 6. Arquitetura da rede MLP utilizada como classificador treinado com os resultados das ativações da VGG-19, Xception e da *Random Forest*.

Analisando os resultados em que o ensemble errou a classificação, foi possível identificar que a *Random Forest* possui muita importância no resultado da classificação. Diversos casos em que a *Random Forest* errou a classificação, o ensemble também errou, desconsiderando os demais classificadores.

Isso ocorre em grande parte pelo fato dos extratores baseados em CNN terem saída com ativação *softmax*, retornando valores decimais entre 0 e 1, enquanto a *Random Forest* possui como saída apenas 0 ou 1, dando a ela, artificialmente, um peso maior que dificulta o aprendizado do classificador MLP.

Consideramos que trabalhos futuros poderão avaliar como suavizar a saída da *Random Forest* ou definir um peso para elas.

3.3.5 Agregador MLP com dados adicionais de endereço

Adicionalmente, um classificador baseado em uma rede MLP foi treinado em uma base que concatena as ativações dos três classificadores escolhidos (VGG-19, Xception, *Random Forest*), além dos campos derivados do CEP e indicador do tipo de logradouro (rua ou avenida).

Tabela 10. Resultados de desempenho dos diversos agregadores, obtidos sobre a base de validação.

Agregador	Acurácia Balanceada	AUC-ROC
Votação - maioria simples	0.88	0.88
Votação – consenso	0.73	0.73
Votação – minoria	0.86	0.86
Soma de ativações	0.90	0.97
Classificador SVM	0.87	0.87
Agregador MLP ☆	0.91	0.97
Agregador MLP com CEP	0.91	0.96
Agregador MLP com CEP e tipo de logradouro	0.89	0.96

3.4 Consolidação dos Resultados com a Base de Validação

Resultados obtidos com a base de validação.

Tabela 11. Resultados consolidados dos melhores classificadores.

Seção	Base	Variação	Acu. Bal.	AUC- ROC
3.1.1	Imagem	CNN Simples - Aprendiz. todos os parâmetros	0.69	0.75
3.1.2.2	Imagem	Transferência de Aprendizado – Camadas Congeladas - Resnet - 50 (*baseline*)	0.82	0.90
3.1.3	Imagem	Transferência de Aprendizado - *Fine Tuning* - VGG-19 (ADAM; lr=0.0001; base de dados aumentada; sem regularização l2)	0.88	0.94
3.2	Endereço	*Random Forest* (n=100)	0.88	0.88
3.3	*Ensemble*	Agregador MLP ☆	0.91	0.97

4 Compreendendo os Classificadores

Para analisar as regiões das imagens que mais influenciaram a decisão das Redes CNN, utilizamos a técnica "*Class Activation Map*"[4] (CAM). A técnica e o código utilizado foram adaptados de https://jacobgil.github.io/deeplearning/class-activation-maps . A influência dos diferentes componentes do CEP na decisão da *Random Forest* foi analisada utilizando a biblioteca "*Tree Interpreter*", cujo funcionamento é detalhado em https://blog.datadive.net/interpreting-random-forests/ .

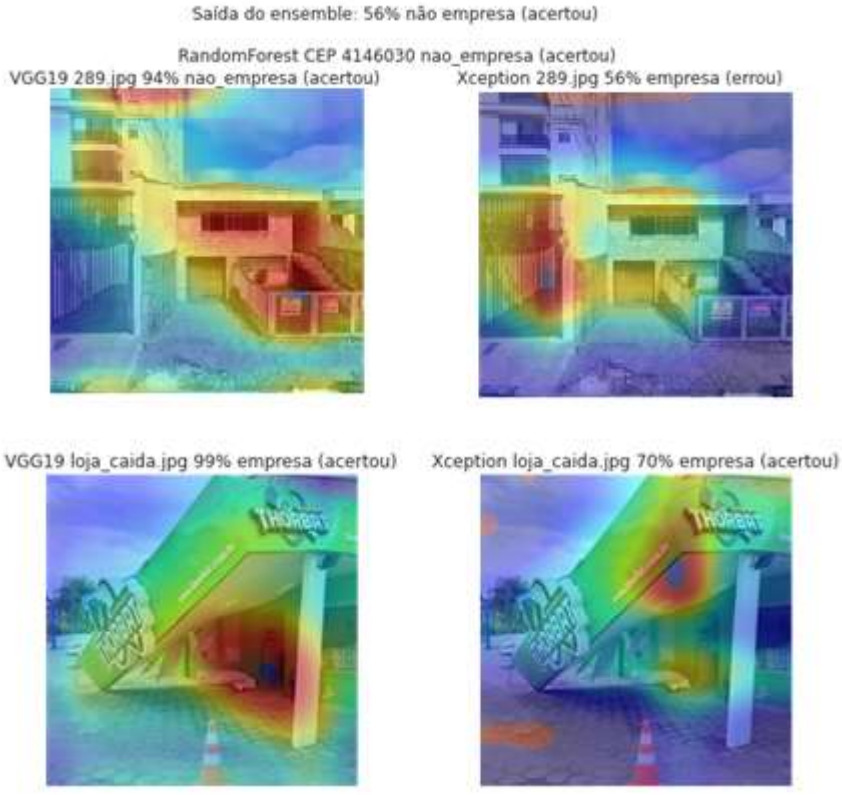


Figura 7. Exemplos de Mapa de ativação.

Nas imagens da primeira linha da Figura 7 note que as redes dedicam mais atenção à porção da fachada do imóvel que corresponde à portões e janelas. O classificador *ensemble* acertou o resultado, unindo os resultados da *Random Forest* (correto), VGG-19 (correto) e Xception (errado).

As imagens da segunda linha da Figura 7 não fazem parte da base de dados fornecida e apresentam alterações atípicas (queda parcial da fachada do estabelecimento). Apesar disso, as redes conseguiram classificar corretamente o local.

[4] Zhou, Bolei, et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Figura 8. Exemplos de Mapas de Ativação. Imagem de cachoeira da classe "empresa".

Na figura 8, a *Random Forest* sobre dados de CEP contribuiu principalmente em casos em que a foto não apresentava elementos visuais comuns à classe. No entanto, percebemos que esse classificador aprendeu vieses dos dados. Por exemplo, todos os CEP com região 1 (interior de São Paulo) são classificados como 'empresa' porque todas as ocorrências de 'não empresa' da base dados são endereços na região 0 (capital).

5 Resultados com a Base de Testes

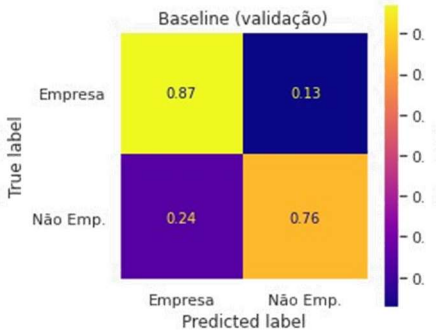
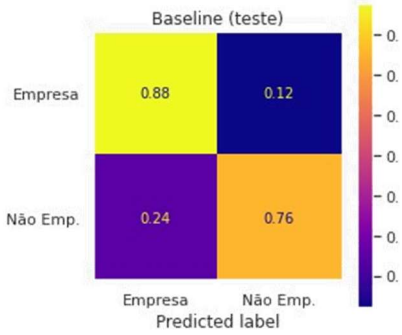
A tabela 12 apresenta os resultados do classificador *baseline*, bem como dos melhores classificadores de cada tipo (avaliados pelos resultados no conjunto de validação) utilizando o conjunto de testes.

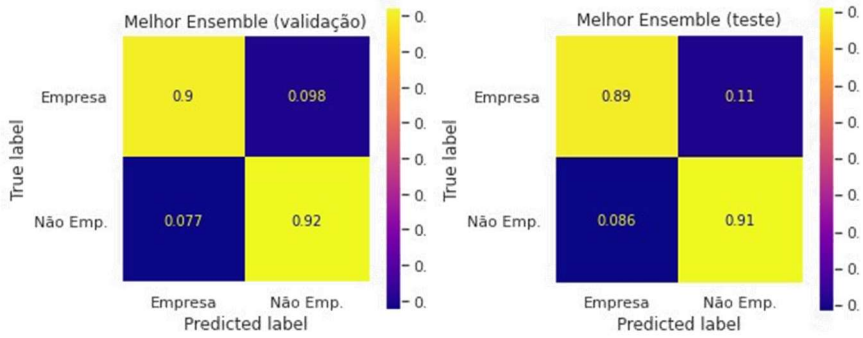
Tabela 12. Resultados obtidos com a base de testes.

	Validação		**Testes**	
Variação	**Acu. Bal.**	**AUC**	**Acu. Bal.**	**AUC**
Transferência de Aprendizado – Camadas Congeladas - Resnet - 50 (baseline)	0.82	0.90	0.82	0.90
Random Forest (n=100)	0.88	0.88	0.87	0.87
VGG-19 (ADAM; lr=0.0001; base de dados aumentada; sem regularização l2)	0.88	0.94	0.87	0.90
Ensemble MLP de VGG-19, Xception, *Random Forest* ☆	0.91	0.97	0.90	0.96

Os resultados obtidos com a base de testes foram muito semelhantes aos obtidos com a base de validação. Em especial, confirmou-se o *ensemble* como o melhor classificador.

Tabela 13. Matrizes de confusão do classificador *baseline* e do melhor classificador.

	Validação	**Testes**
Transferência de Aprendizado – Camadas Congeladas – Resnet – 50 (*baseline*)		

Ensemble MLP de VGG-19, Xception, *Random Forest*	

No melhor classificador houve uma expressiva redução dos casos de predição equivocada da classe “não empresa” em comparação com o *baseline*.

6 Conclusões

Verificamos neste trabalho que a transferência de aprendizado de redes pré-treinadas pode ser aplicada com sucesso no domínio do problema proposto. Muito embora as redes pré-treinadas disponíveis no pacote Keras tenham sido treinadas com imagens de outros domínios (ImageNet), elas foram um excelente ponto de partida para o treinamento dos parâmetros.

Nas análises realizadas com a base de validação, identificamos que o melhor classificador de imagens foi baseado na VGG-19. A partir dessa constatação, avaliamos diversas configurações de hiperparâmetros das redes, como otimizador, taxa de aprendizado, balanceamento e regularização. No entanto, essas alterações tiveram pouco impacto nos resultados, medidos tanto por acurácia balanceada quanto por AUC-ROC.

Além de refinar os classificadores, buscamos aumentar a quantidade de imagens da base de treino. Como a base original inclui os endereços e imagens provenientes do Google Street View, foi possível utilizar esse mesmo serviço para trazer imagens ligeiramente diferentes dos mesmos endereços. Além do Google Street View, imagens de satélites poderiam ter sido utilizadas, além de fotos de outros provedores de mapas, como o Microsoft Bing Maps.

Como esperado, as imagens adicionais aumentaram em muito o esforço computacional para treinar as redes e tiveram impacto positivo na acurácia balanceada dos classificadores. O aumento foi da ordem de 1%. Consideramos que outros analistas visando utilizar esta abordagem devam avaliar se o custo computacional justifica o ligeiro aumento em acurácia.

Paralelamente ao refinamento dos classificadores de imagens, utilizamos um classificador *Random Forest* para os dados tabulares. Apesar de utilizar somente dados derivados do campo CEP, a *Random Forest* apresentou acurácia balanceada inesperadamente elevada, comparável à dos classificadores de imagens.

Uma vez avaliados diversos classificadores isoladamente, utilizamos algumas abordagens de *ensembles*, obtendo um ganho substancial com relação ao melhor classificador individual (rede VGG-19 com dados aumentados) ao combinar os classificadores de imagens e de dados tabulares utilizando um classificador MLP para agregar os resultados.

Para entender melhor como as redes estariam "interpretando" as imagens, foi utilizada a técnica de *Class Activation Maps* (CAM), que, juntamente com o uso de uma biblioteca de avaliação de *Random Forest*, permitiu análise dos resultados e identificação de pontos de melhoria.

Por fim, acreditamos que todo o trabalho realizado na criação de um classificador para identificar "empresas" e "não empresas" tenha sido bem-sucedido. Além do aumento da acurácia balanceada sobre a base de testes de 87% (nosso baseline) para 90%, obtivemos um melhor entendimento da base de dados e, principalmente, ideias e questionamentos que serão muito úteis para a aplicação do conhecimento adquirido ao longo do curso em outros problemas da nossa área de atuação.