

# **Análise de Diferentes Algoritmos Semissupervisionados para *Data Augmentation***

**Willian Dihanster Gomes de Oliveira**

**Orientadora: Lilian Berton, Co-orientador: Otávio A. B. Penatti**

**Instituto de Ciência e Tecnologia - Universidade Federal de São Paulo**

**04 de Junho de 2019**

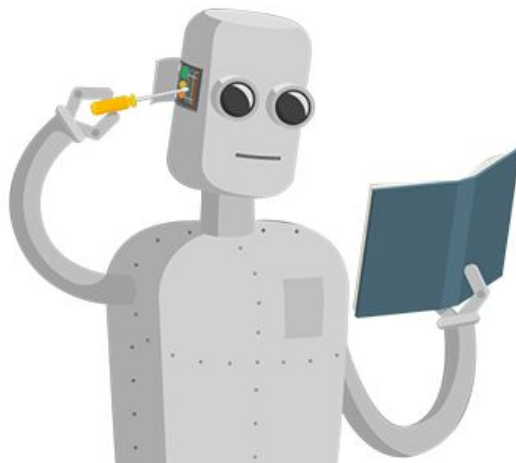
**São José dos Campos - SP**

**V Congresso Acadêmico**

**“UNIFESP 25 anos: Universidade Pública, Conhecimento Público.”**

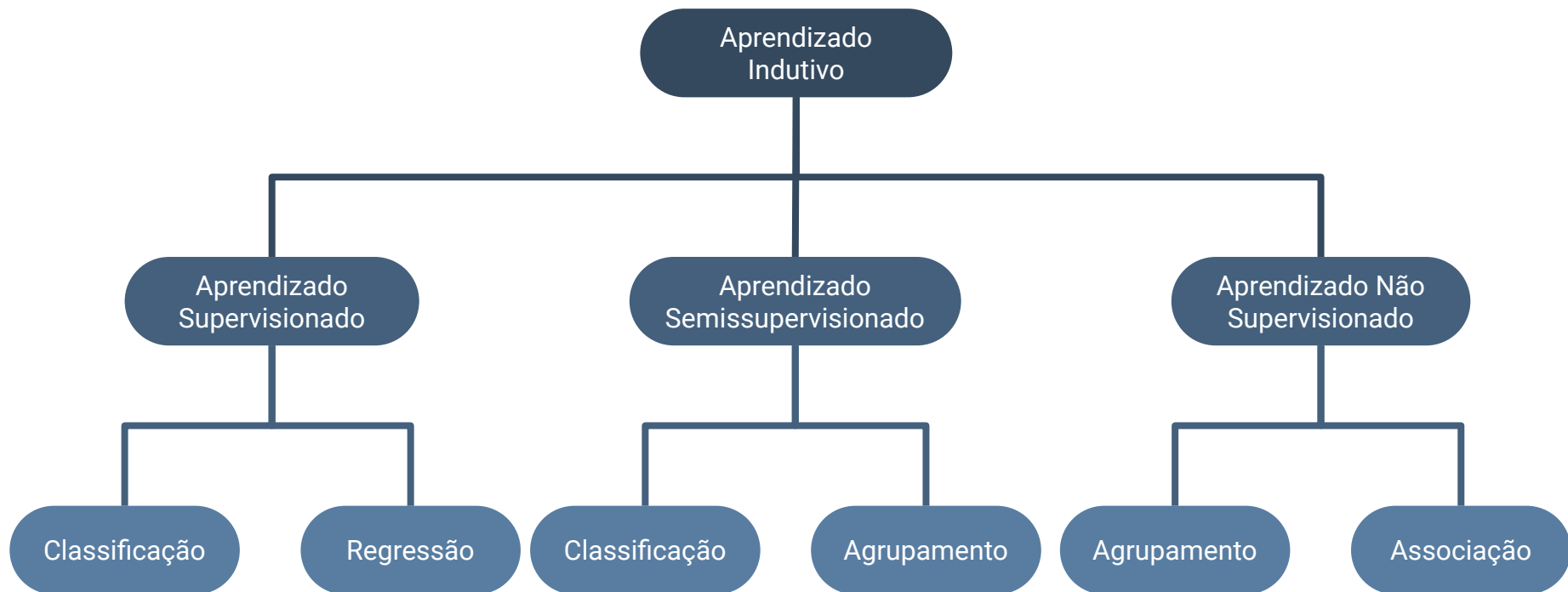
# Introdução

- Aprendizado de Máquina (AM):
  - Aprender a partir de exemplos.
  - Melhorar automaticamente com a experiência. [1]
  - Aprendizado indutivo: Usa indução, que é a forma de inferência lógica que permite obter conclusões genéricas sobre um conjunto particular de exemplos. [2]



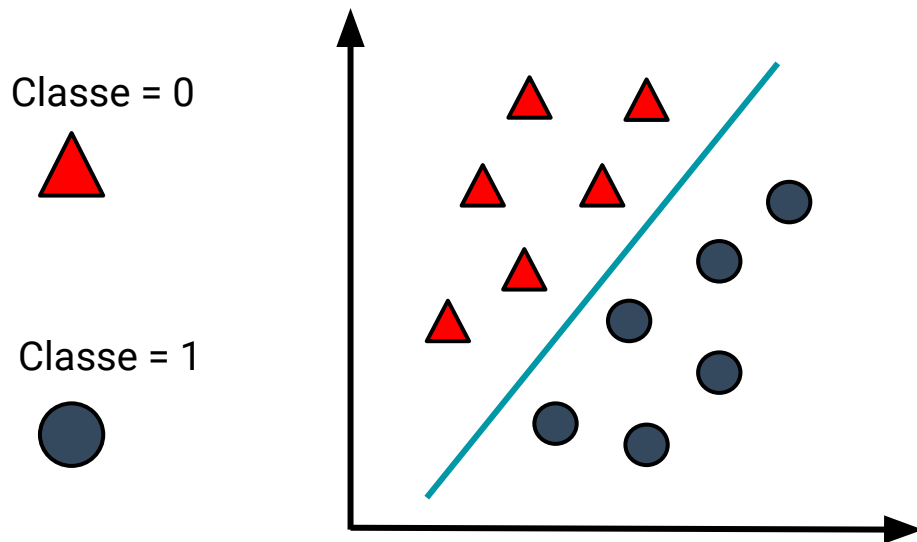
Robô ajustando seus parafusos para aprender. Disponível em:  
<http://fiori-implementation.com/blog/make-clean-breast-machine-learning/>

# Hierarquia de Aprendizado Indutivo



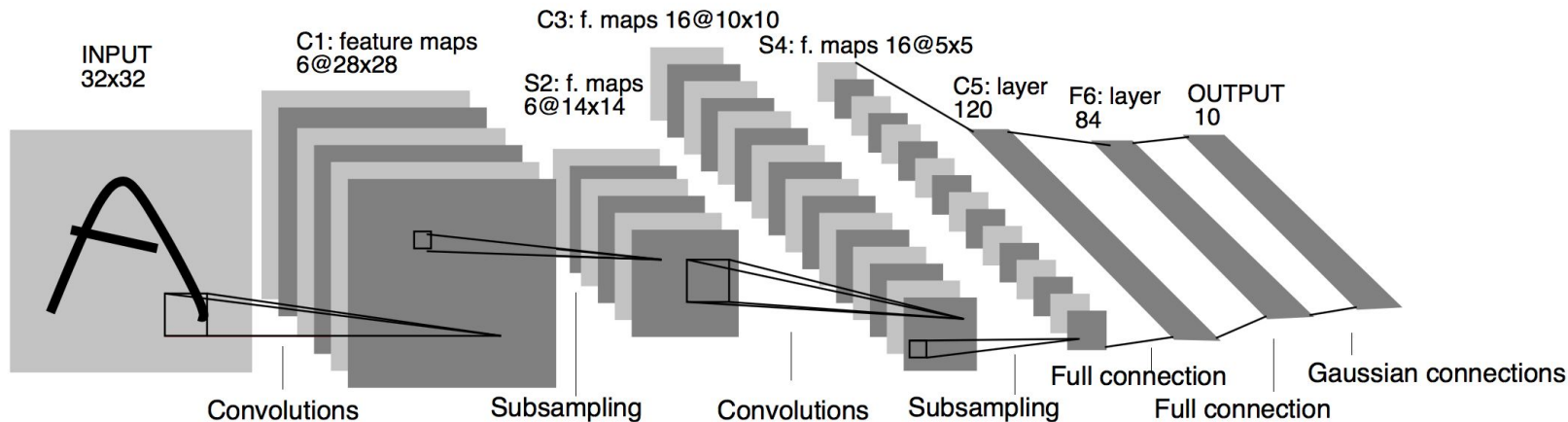
# Classificação

- Rótulo ou classe:
  - É um atributo especial que separa os exemplos/atributo que queremos prever.
- O objetivo da classificação é modelar um estimador que para cada exemplo ainda não rotulado, atribua esse exemplo a uma das possíveis classes.



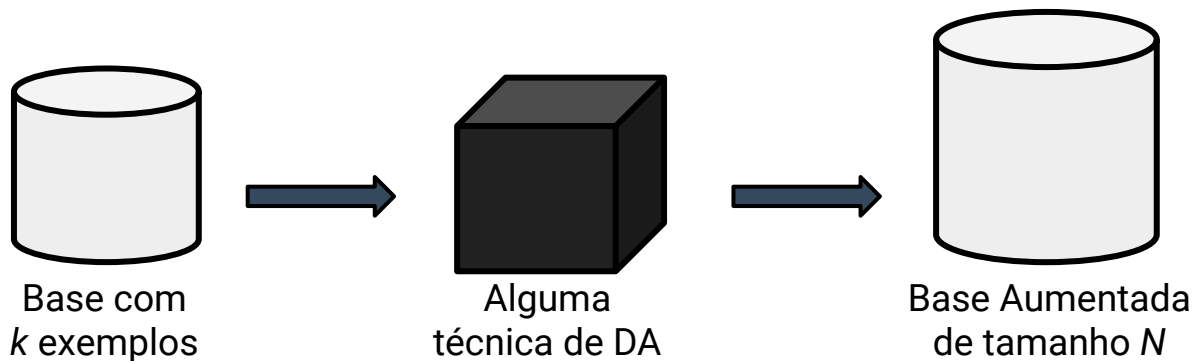
# Classificação e Redes Neurais Convolucionais

- Classificação de imagens: reconhecimento facial/objetos, detecção de tumores, etc.
- Extração de características: Descrever as imagens em vetores de atributos.
- Redes Neurais Convolucionais:
  - Convolução: Gera o mapa (matriz) de *features*.
  - *Pooling*: Reduz o tamanho do mapa de *features*.
  - Camadas Totalmente Conectadas e de Saída: Fase de classificação.



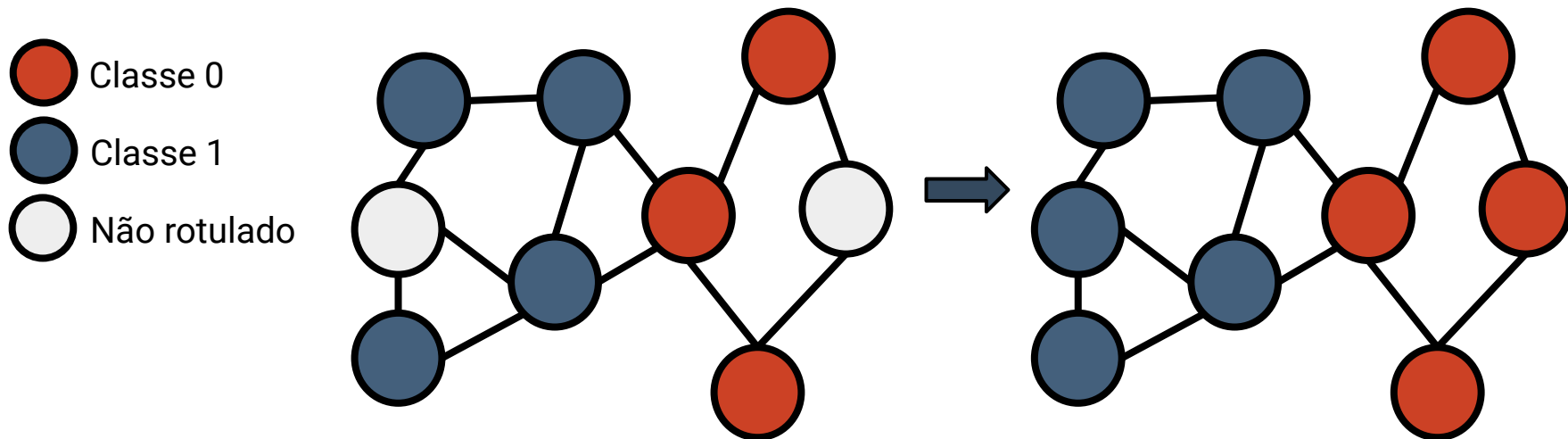
# Data Augmentation

- Aprendizado Supervisionado: quanto mais dados (rotulados), maior a capacidade de predição dos algoritmos.
- Mas, dados rotulados são custosos e difíceis de se obter.
- *Data Augmentation*:
  - Técnicas computacionais para aumentar bases de dados.



# Algoritmos Semissupervisionado Baseado em Grafos

- Pode ser utilizado quando há dados rotulados e não rotulados.
- Transdutivo:
  - Classifica apenas os dados atuais, e não gera uma função para novos dados.
- Algoritmos de propagação de rótulos:
  - A partir da informação de rótulo de alguns dados, inferir o rótulo do resto.

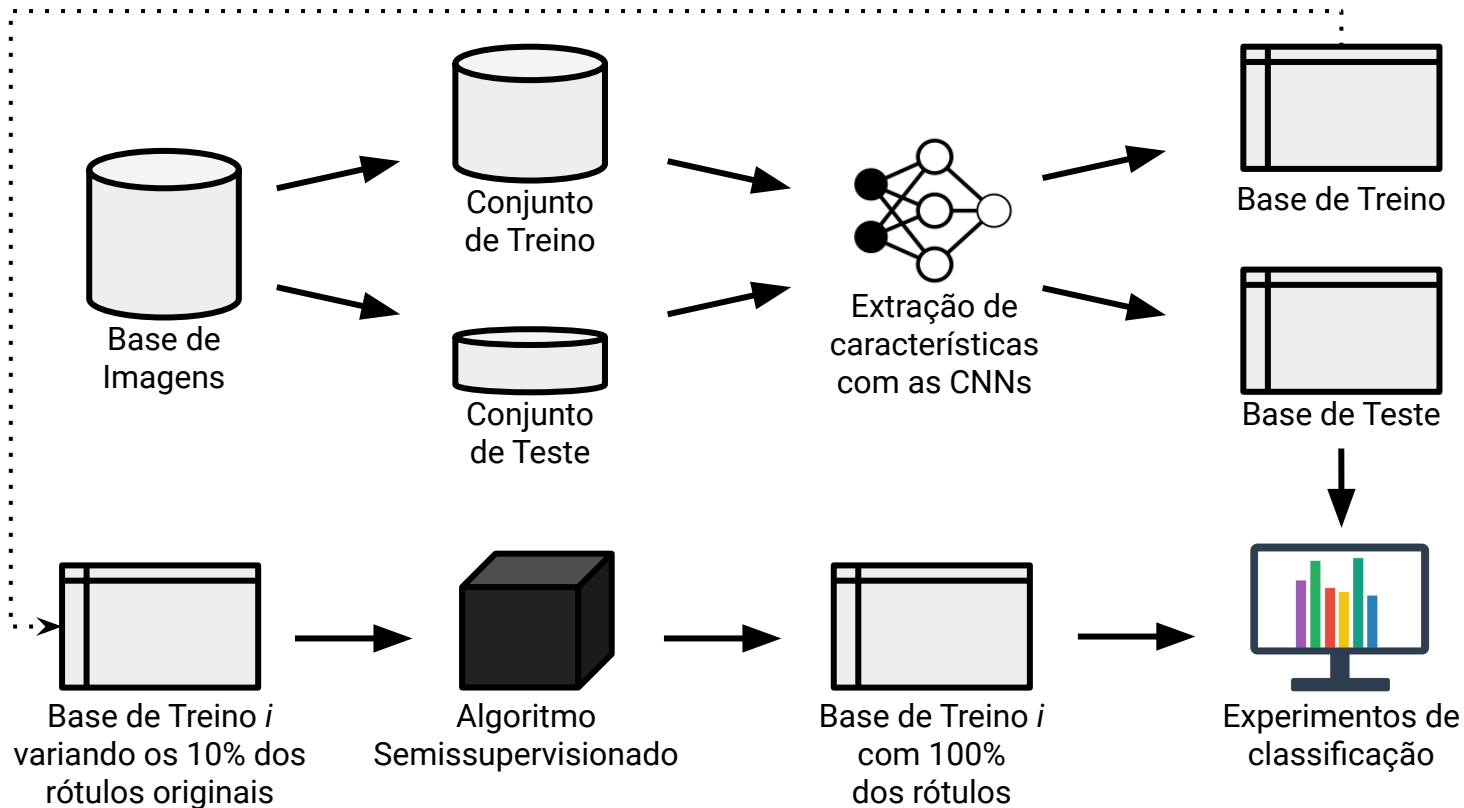


# Objetivos

- Estudo de algoritmos semissupervisionados.
  - Simulação e comparação dos algoritmos semissupervisionados.
- Estudo da Redes Neurais Convolucionais (CNNs).
  - Uso e avaliação das CNNs como extratores de características.



# Metodologia



# Base 1: *Brazilian Coffee Scenes Dataset*

- A base de dados [3] é composta por imagens de satélite de plantações de café em cidades de Minas Gerais.
- Cada imagem possui dimensão 64x64 e classificada manualmente por pesquisadores agrícolas como "coffee" ou "non coffee".
- 2876 imagens, 50% de exemplos rotulados como "coffee" e 50% "non coffee".
  - 2010 exemplos para treino e 766 para teste.



Exemplos rotulados como "Coffee".



Exemplos rotulados como "Non-coffee".

## Base 2: *Dogs vs Cats*

- Essa base é composta por imagens de cachorros e gatos, para um competição do Kaggle [4].
- Todas imagens possuem dimensão variada.
- 10.000 exemplos para treino (50% de cada classe) e 2.000 para teste.



Exemplos rotulados como “Dog”.



Exemplos rotulados como “Cat”.

## Base 3: *MNIST* (3-8)

- Neste caso, um subconjunto da base *MNIST* [5] com exemplos do dígito 3 e dígito 8.
- Cada imagem possui dimensão 64x64.
- 5.600 exemplos para treino (50% de cada classe) e 2.400 exemplo para teste.



Exemplos rotulados como “3”.



Exemplos rotulados como “8”.

# Ferramentas

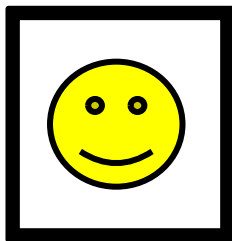
- Python 3.6
- TensorFlow, Keras
- NumPy, Pandas e scikit-learn



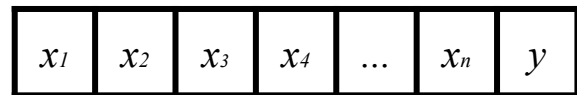
# Extração de Características

## Redes Neurais Convolucionais:

- VGG16 e VGG19
- ResNet50
- Xception



Imagem



Vetor de atributos

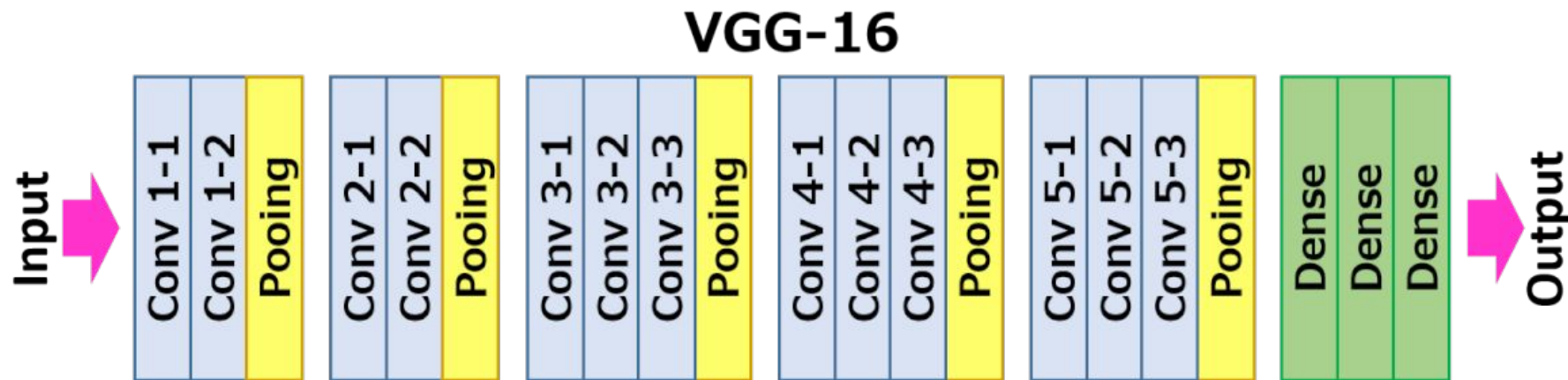
## Transferência de Aprendizado:

- Modelo treinado para uma tarefa é reusado em outra tarefa.
- Redes pré-treinadas com Imagenet (+14 milhões) [6].
- Sem treinar últimas camadas (*finetuning*).
  - Economia de tempo de treinamento e processamento.



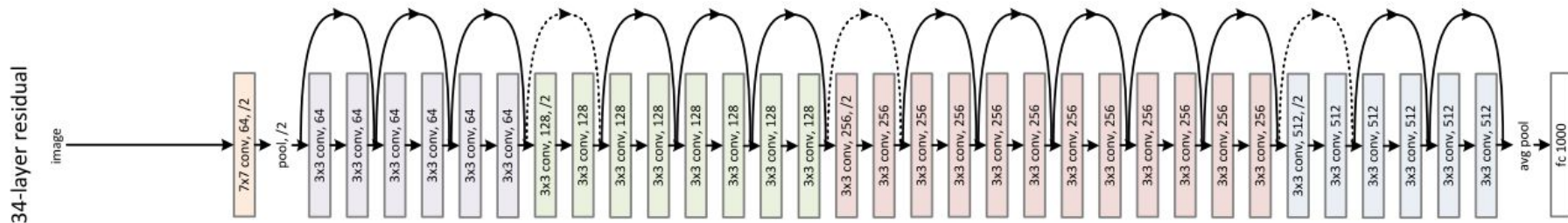
# VGG16 e VGG19

- A VGG16 e VGG19 [7] possuem 16 e 19 camadas de convolução respectivamente.
- Matrizes de filtros 3x3 e  $stride = 1$ .
- 5 camadas *max pooling* 2x2 e  $stride = 2$ .
- 3 camadas totalmente conectadas.
- Vetor de *features* da última camada (antes da classificação) é 1x4096.



# ResNet50

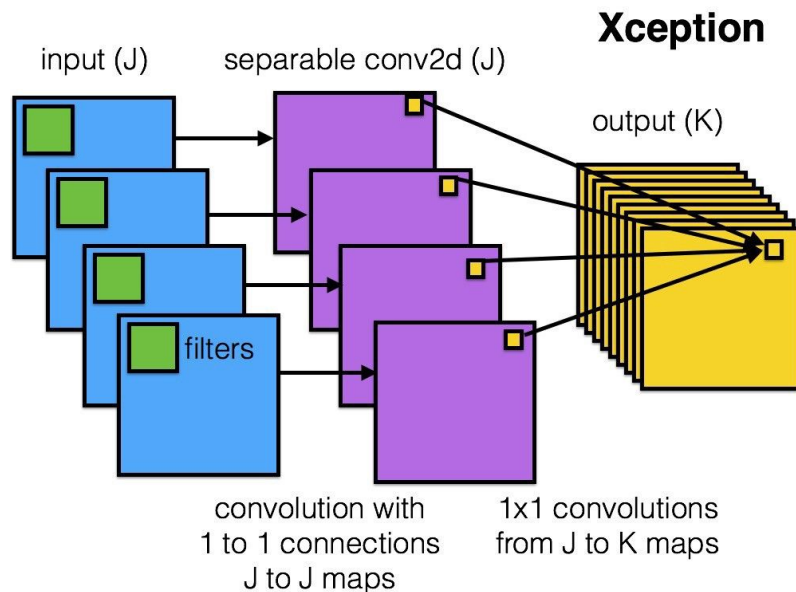
- ResNet50 [8] possui 50 camadas de convolução.
- Muitas camadas:
  - Complexidade, tempo e degradação da acurácia.
  - Solução: Técnica de *Residual Learning*.
- Convolução inicial de 7x7 e um *max pooling* com *stride* = 2.
- Seguido por 49 Convoluções (1x1 e 3x3), um *global avg pooling* e uma camada totalmente conectada.
- Vetor de *features* da última camada é 1x2048.





# Xception

- *Xception* ou *Extreme Inception* [9] possui 36 camadas de convolução.
- Separada em módulos e usa técnica de *Residual Learning*.
- Convoluções 3x3 (1x1 nas camadas residuais) e *stride* = 2
- Vetor de *features* da última camada é 1x4096.



# Algoritmos Semissupervisionados

## Algoritmos:

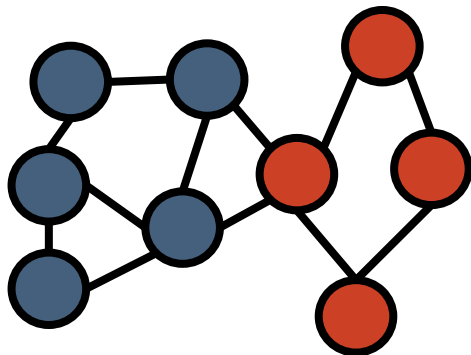
- *Label Propagation* (LP) [10]
- *Local and Global Consistency* (LGC) [11]
- *Gaussian Fields and Harmonic Functions* (GFHF) [12]
- *OMNI-Prop* (OMNI) [13]

## Construção do Grafo:

- Grafo  $k$ NN - Um dado nó é conectado aos seus  $k$  nós mais próximos.

## Modelagem:

- **Nós:** os exemplos da base.
- **Arestas:** conexão baseado na semelhança dos exemplos.
- Apenas 10% dos nós com informação dos rótulos.



# Label Propagation

- Os rótulos são propagados para os vizinhos de acordo com a proximidade.
  - Inclusive os exemplos já rotulados podem ser alterados.

---

**Algoritmo 2.6:** Propagação de rótulos (Zhu & Ghahramani, 2002)

---

- 1 Calcule a matriz de afinidade  $W$  utilizando a Equação (2.30) ;
  - 2 Calcule a matriz diagonal  $D$  utilizando  $D_{ii} \leftarrow \sum_j W_{ij}$  ;
  - 3 Inicialize  $\hat{Y}^{(0)} \leftarrow (y_1, y_2, \dots, y_l, 0, 0, \dots, 0)$  ;
  - 4 **repita**
    - 5  $\hat{Y}^{(t+1)} \leftarrow D^{-1}W\hat{Y}^{(t)}$  ;
    - 6  $\hat{Y}_l^{t+1} = Y_l$  ;
  - 7 **até** a convergência para  $\hat{Y}^{(\infty)}$ ;
  - 8 Rotule a amostra  $x_i$ , usando o sinal de  $\hat{y}_i^\infty$  ;
-

# Local and Global Consistency

- Se baseia na ideia de Consistência Global e Local:
  - Global: Pontos de mesma estrutura são prováveis de terem mesmo rótulo.
  - Local: Pontos próximos são prováveis de possuírem o mesmo rótulo.

---

**Algoritmo 2.4** Consistência local e global (Zhou et al., 2004)

---

- 1: Construa uma matriz de afinidade  $\mathbf{W}$ , onde  $w_{ii} \leftarrow 0$  e  $w_{ij}$  é dado pela Eq. 2.13 para  $i \neq j$ ;
  - 2: Calcule a matriz diagonal  $\mathbf{D}$ , onde  $d_{ii} \leftarrow \sum_j w_{ij}$  e  $d_{ij} \leftarrow 0$  para  $i \neq j$ ;
  - 3: Calcule o Laplaciano normalizado da rede,  $\mathcal{L} \leftarrow \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$ ;
  - 4: Escolha um parâmetro  $\alpha \in [0, 1)$ ;
  - 5: Inicialize  $f_j(0) \leftarrow y_j$  para todo  $j \in X_l$  e  $f_i(0) \leftarrow 0$  para todo  $i \in X_u$ ;
  - 6: **repita**
  - 7:    $f(t+1) \leftarrow \alpha \mathcal{L} f(t) + (1 - \alpha) Y$ ;
  - 8: **até que**  $f$  convirja;
  - 9: Rotule cada item de dado usando o sinal de  $f(\infty)$ .
-

# Gaussian Fields and Harmonic Functions

- O algoritmo utiliza funções harmônicas para rotular os dados:
  - Um rótulo atribuído a um nó será dado pela média ponderada dos seus vizinhos.
  - Gerando rotulações mais suaves.

---

**Algoritmo 2.3** Campos Gaussianos e funções harmônicas (Zhu & Ghahramani, 2002; Zhu et al., 2003)

---

- 1: Construa uma matriz de afinidade  $\mathbf{W}$ , onde  $w_{ii} \leftarrow 0$  e  $w_{ij}$  é dado pela Eq. 2.13 para  $i \neq j$ ;
  - 2: Calcule a matriz diagonal  $\mathbf{D}$ , onde  $d_{ii} \leftarrow \sum_j w_{ij}$  e  $d_{ij} \leftarrow 0$  para  $i \neq j$ ;
  - 3: Inicialize  $f_j(0) \leftarrow y_j$  para todo  $j \in X_l$  e  $f_i(0) \leftarrow 0$  para todo  $i \in X_u$ ;
  - 4: **repita**
  - 5:    $f(t+1) \leftarrow \mathbf{D}^{-1}\mathbf{W}f(t)$ ;
  - 6:    $f_j(t+1) \leftarrow f_j(t)$ ;
  - 7: **até que**  $f$  convirja;
  - 8: Rotule cada item de dado usando o sinal de  $f_i(\infty)$ .
-

# OMNI-PROP

- Se os vizinhos de um nó possuem um rótulo, esse nó também tem esse rótulo.
- Não parte do pressuposto da rede ser homofílica (nós semelhantes tendem a ter conexão) ou heterofílica (nós semelhantes tendem a não ter uma conexão).
- Considera se as evidências são suficientes antes de rotular um exemplo.

---

## Algorithm 1 Iterative Algorithm

---

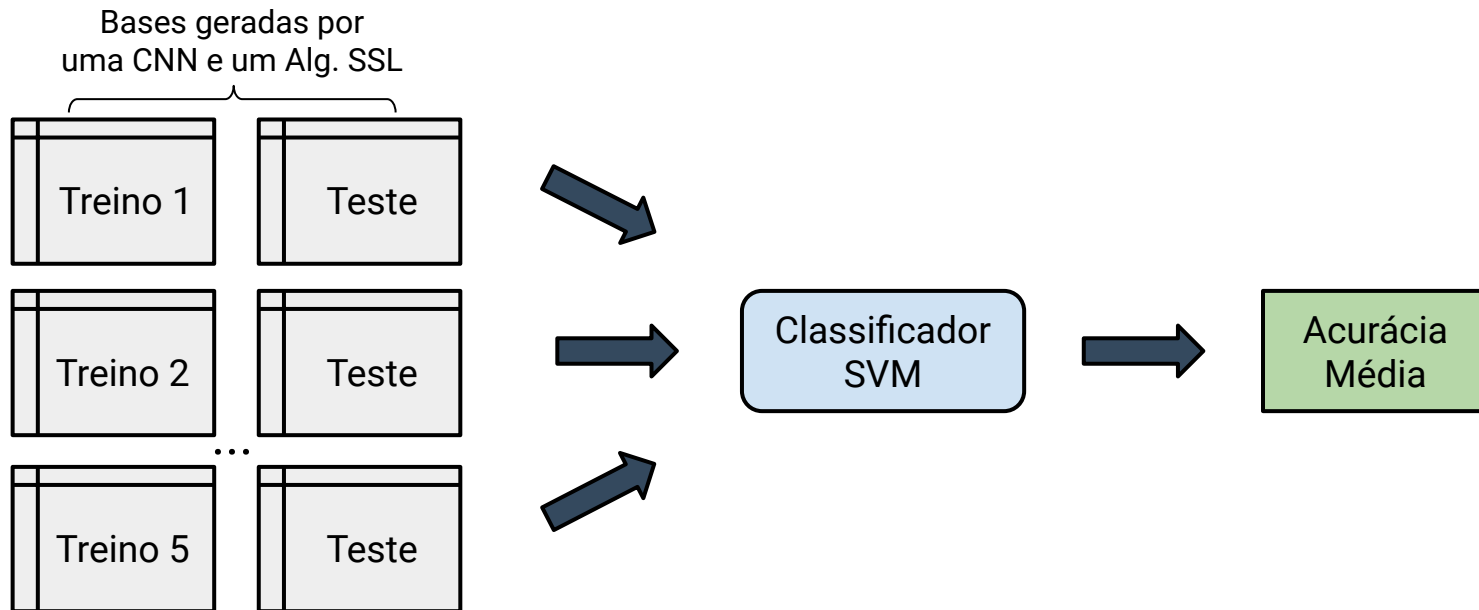
**Require:** explicit labels  $Y_L$ , adjacency matrix  $A$ , parameter  $\lambda$

```
1:  $\mathbf{b} \leftarrow \text{uniform}()$ 
2:  $S^0 \leftarrow \text{initializeS}(Y_L)$ 
3:  $T^0 \leftarrow \text{initializeT}()$ 
4:  $k \leftarrow 0$ 
5: repeat
6:    $S_U^{k+1} \leftarrow (D_U + \lambda I)^{-1} (A_U T^k + \lambda \mathbf{1}_U \mathbf{b}^T)$ 
7:    $T^{k+1} \leftarrow (F + \lambda I)^{-1} (A^T S^k + \lambda \mathbf{1}_N \mathbf{b}^T)$ 
8:    $k \leftarrow k + 1$ 
9: until error between  $S_U^{k+1}$  and  $S_U^k$  becomes sufficiently small
10: return  $S_U^k$  and  $T^k$ 
```

---

# Validação dos Dados e Algoritmos

- Para avaliar os resultados obtidos, a medida utilizada foi a Acurácia.
  - Média de acertos.
- Processo repetido para cada CNN e para cada algoritmo SSL.



# Resultados

- *Brazilian Coffee Scenes Dataset*

	ResNet50	VGG16	VGG19	Xception	Média
Original (5-fold)	90,76	83,26	86,49	77,83	84,59
LP	79,79	68,59	72,01	66,79	71,80
LGC	81,33	70,92	68,15	71,62	73,01
GFHF	<b>84,04</b>	72,95	76,47	73,42	<b>76,72</b>
OMNI	82,54	66,93	68,82	72,84	72,78



# Resultados

- *Dogs vs Cats*

	ResNet50	VGG16	VGG19	Xception	Média
Original (5-fold)	98,60	98,55	98,55	69,95	91,41
LP	98,07	97,46	97,55	55,63	87,18
LGC	<b>98,46</b>	98,19	98,06	60,82	88,88
GFHF	98,37	98,18	97,87	62,33	<b>89,19</b>
OMNI	98,36	98,19	97,82	62,26	89,16

# Resultados

- *MNIST (3-8)*

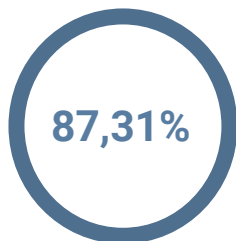
	ResNet50	VGG16	VGG19	Xception	Média
Original (5-fold)	98,60	98,55	98,55	97,17	98,22
LP	97,20	96,59	96,95	75,87	91,65
LGC	84,45	86,49	86,89	86,86	86,17
GFHF	<b>98,38</b>	97,69	97,47	91,23	<b>96,19</b>
OMNI	98,23	97,79	97,37	89,66	95,76

# Resultados

- Avaliação das Redes Neurais Convolucionais
  - Média de todos os experimentos que envolvem tal rede.



ResNet50



VGG16



VGG19

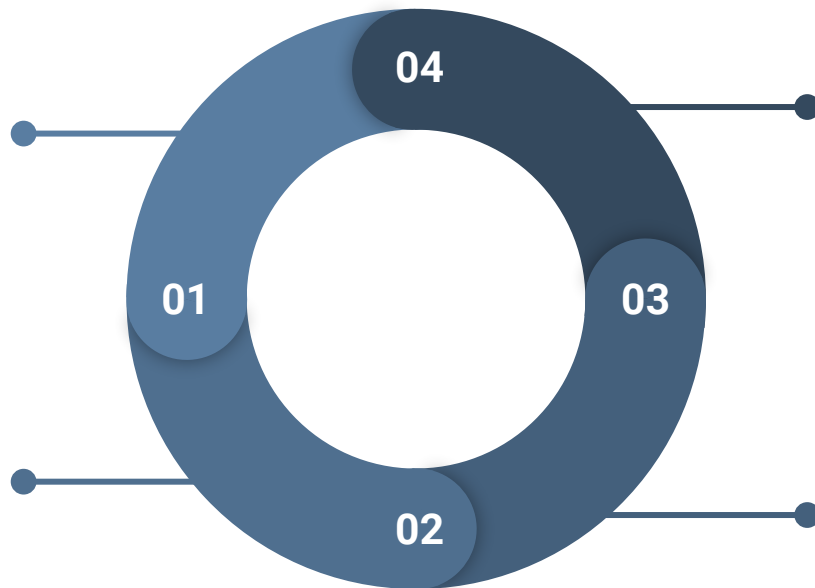


Xception

# Resultados

**Label Propagation**  
Média de 83,54%

**Local and Global  
Consistency**  
Média de 86,18%



**Gaussian Fields and  
Harmonic Functions**  
Média de 87,37%

**OMNI-Prop**  
Média de 85,91%

# Conclusões

- Em aplicação onde se tem poucos exemplos rotulados e muitos exemplos não rotulados, pode-se aplicar um algoritmo semissupervisionado para aumentar a quantidade de exemplos rotulados e, conseqüentemente, obter melhores resultados.
  - Todos algoritmos obtiveram bom desempenho.
- Redes Neurais Convolucionais podem ser aplicadas como bons extratores de características.
  - Mesmo com transferência de aprendizado (e sem *finetuning*) se saíram bem.
  - Destaque para a ResNet50:
    - Rápida, menos pesada e melhores resultados em geral.

# Agradecimentos

- Esse projeto foi apoiado e financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.



# Referências bibliográficas

- [1] Michalski, Ryszard S., et al., eds. *Machine learning: A multistrategy approach*. Vol. 4. Morgan Kaufmann, 1994.
- [2] MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, v. 1, n. 1, p. 32, 2003.
- [3] Penatti, Otávio AB, Keiller Nogueira, and Jefersson A. dos Santos. "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015 IEEE Conference on. IEEE, 2015.
- [4] KAGGLE. Dogs vs. Cats. 2015. Disponível em: <<https://www.kaggle.com/c/dogs-vs-cats-/data>>.
- [5] Lecun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219.
- [6] DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. 2009 IEEE conference on computer vision and pattern recognition. [S.l.], 2009. p. 248–255.
- [7] SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.
- [8] HE, K. et al. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun 2016.
- [9] CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1251–1258.
- [10] Zhu, Xiaojin, and Zoubin Ghahramani. *Learning from labeled and unlabeled data with label propagation*. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.
- [11] Zhou, Denny, et al. "Learning with local and global consistency." *Advances in neural information processing systems*. 2004.
- [12] Zhu, Xiaojin, Zoubin Ghahramani, and John D. Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions." *Proceedings of the 20th International conference on Machine learning (ICML-03)*. 2003.
- [13] Yamaguchi, Yuto, Christos Faloutsos, and Hiroyuki Kitagawa. "Omni-prop: Seamless node classification on arbitrary label correlation." *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

Obrigado!

