

Projet HMIN339M

Reconnaissance visuelle de plantes

Titouan Lorieul

Zenith
LIRMM – Inria
Université de Montpellier

17 octobre 2019



Outline

Présentation du problème

(Très) courte introduction à la reconnaissance d'images

Détails du projet

Outline

Présentation du problème

(Très) courte introduction à la reconnaissance d'images

Détails du projet

Jeu(x) de données

But: reconnaissance d'espèces de plantes à partir de photos



(a) *Ficaria verna* Huds.



(b) *Cichorium intybus* L.



(c) *Centaurea jacea* L.



(d) *Tulipa sylvestris* L.

Jeu de départ: 3474 images appartenant à 50 espèces différentes

Remarques

Objectif du projet

Le but est de vous faire travailler pas seulement l'aspect technique mais aussi de vous faire réfléchir au problème en lui-même.

C'est à vous d'analyser le jeu de données pour en comprendre les *caractéristiques* avant de *formaliser* le problème !

Données de test

Les images sont regroupées par observations, chaque observation correspondant à un même specimen. Les jeux de train et de test n'ont pas d'images appartenant à une même observation.

Le jeu de test vous sera fourni ultérieurement.

Autres jeux de données

Potentiellement, en fonction de l'avancement, d'autres jeux de données plus conséquents pourront être mis à disposition.

Outline

Présentation du problème

(Très) courte introduction à la reconnaissance d'images

Détails du projet

Data representation: images

Question: what is represented in this image?

```
[[ 0  0 80 208 144 16  0  0]
 [ 0  0 208 240 160 240 80  0]
 [ 0 48 240 32  0 176 128  0]
 [ 0 64 192  0  0 128 128  0]
 [ 0 80 128  0  0 144 128  0]
 [ 0 64 176  0 16 192 112  0]
 [ 0 32 224 80 160 192  0  0]
 [ 0  0 96 208 160  0  0  0]]
```

Data representation: images

Question: what is represented in this image?

```
[[ 0  0 80 208 144 16  0  0]  
[ 0  0 208 240 160 240 80  0]  
[ 0 48 240 32  0 176 128  0]  
[ 0 64 192  0  0 128 128  0]  
[ 0 80 128  0  0 144 128  0]  
[ 0 64 176  0 16 192 112  0]  
[ 0 32 224 80 160 192  0  0]  
[ 0  0 96 208 160  0  0  0]]
```



Data representation: images

And now?

```
[[255 252 252 251 250 249 249 248 249 247 247 246 238 217 215 220 216 207 200 194 188 182 178 179 183 189 195 193 189 182 172 168]
[255 255 254 253 252 253 253 250 251 250 249 247 244 234 224 209 188 184 186 183 180 176 174 178 180 183 190 189 186 181 173 171]
[255 252 251 251 251 250 249 247 242 231 221 209 193 181 177 174 172 174 179 179 176 172 168 171 176 177 182 182 181 178 172 171]
[255 252 250 248 247 241 229 211 200 184 169 127 94 77 80 91 126 162 162 172 173 170 166 168 173 174 178 180 181 179 176 182]
[254 243 238 214 203 193 163 140 133 155 181 117 50 31 27 35 54 101 141 157 189 199 197 194 186 173 164 161 164 167 179 195]
[254 208 192 209 192 179 157 119 103 140 182 174 76 38 27 27 32 39 84 145 177 199 209 215 196 170 156 156 162 167 175 182]
[244 198 215 213 173 201 185 127 93 105 145 176 119 43 28 25 27 29 35 94 146 115 158 185 178 180 184 190 194 194 191 188]
[230 187 188 159 147 190 189 143 93 73 99 145 148 54 30 28 25 28 24 29 89 111 180 211 195 192 191 186 182 182 183 185]
[211 181 148 137 139 161 176 150 109 67 64 99 144 100 32 31 25 27 48 61 99 160 217 222 208 188 175 174 176 184 194 206]
[192 159 148 157 157 149 146 145 121 73 56 64 99 105 43 38 45 74 118 153 169 163 176 195 202 203 194 190 205 214 221 229]
[222 135 124 152 163 153 127 112 118 80 60 55 64 80 88 100 128 156 145 116 100 107 135 162 168 175 184 176 215 237 233 233]
[207 158 143 144 140 129 107 82 87 85 90 99 109 112 115 121 131 125 105 75 89 105 115 119 114 114 129 140 195 237 233 234]
[181 145 145 157 145 133 114 91 85 92 109 121 136 134 108 82 77 73 71 71 78 79 76 78 76 70 75 99 171 219 220 223]
[180 184 91 114 131 144 144 142 134 116 118 101 107 136 147 120 88 73 70 70 71 74 76 73 68 68 67 70 136 201 204 210]
[193 90 59 72 83 104 131 153 158 150 139 185 85 94 141 141 104 86 73 69 69 70 72 73 69 64 61 55 105 165 187 213]
[215 111 72 69 65 65 78 100 119 139 137 110 87 79 105 163 141 83 80 82 70 70 71 68 64 63 42 37 82 123 177 227]
[227 149 108 68 68 65 63 66 69 88 117 95 97 104 82 141 174 157 111 114 111 71 68 59 36 62 56 48 73 119 179 203]
[219 176 138 80 79 79 76 71 66 66 68 38 62 111 108 104 199 227 183 125 151 124 70 55 34 53 67 63 70 108 143 182]
[218 194 153 86 85 86 84 82 78 75 49 29 39 70 96 93 147 227 202 152 132 147 93 67 62 64 66 66 70 73 92 182]
[225 208 161 84 79 83 82 80 79 82 49 45 75 73 79 87 95 182 188 124 95 105 107 74 69 67 67 65 56 44 92 197]
[225 208 156 80 69 78 81 79 78 79 48 79 116 98 78 82 84 128 143 84 75 73 82 72 67 65 61 48 32 40 121 211]
[221 209 133 49 51 69 80 81 77 77 53 110 102 99 72 91 80 87 104 79 70 67 65 66 60 48 37 26 23 46 143 221]
[224 219 169 99 92 83 71 86 86 78 63 137 134 108 72 100 83 75 96 77 64 58 63 58 34 27 24 26 33 62 154 213]
[230 226 228 206 193 168 123 88 101 96 70 130 150 154 117 76 87 89 82 64 39 35 64 48 22 24 31 42 56 113 180 206]
[235 233 233 238 236 226 208 164 113 103 76 115 134 151 83 66 76 67 36 52 23 36 72 34 29 41 54 62 91 165 200 214]
[238 237 237 238 236 232 227 223 193 125 59 128 150 155 68 75 73 53 47 64 37 57 69 45 57 63 70 98 154 197 211 220]
[243 240 246 240 241 238 233 230 227 203 118 121 158 116 108 84 79 69 80 86 69 69 65 66 74 93 124 164 197 210 215 222]
[247 243 243 244 245 243 241 238 234 233 197 116 156 139 136 73 79 82 102 96 76 77 85 104 133 163 187 203 210 215 217 222]
[251 247 247 248 249 248 245 244 244 241 239 163 116 152 90 43 70 85 109 113 112 131 156 179 197 207 211 213 215 217 217 220]
[253 250 250 250 250 248 247 247 247 246 222 140 181 74 92 130 150 165 174 184 196 207 217 218 217 214 214 214 215 214 216]
[255 253 253 253 252 251 251 250 250 250 249 247 228 199 191 206 218 222 229 230 226 223 222 226 227 223 220 218 215 215 217]
[253 250 249 249 247 247 247 246 246 246 249 248 246 242 241 241 239 232 225 224 229 230 225 221 220 216 215 214 217]]
```

Data representation: images

And now?

```
[[255 252 252 251 250 249 249 248 249 247 247 246 238 217 215 220 216 207 200 194 188 182 178 179 183 189 195 193 189 182 172 168]
[255 255 254 253 252 253 253 250 251 250 249 247 244 234 224 209 188 184 186 183 180 176 174 178 180 183 190 189 186 181 173 171]
[255 252 251 251 251 250 249 247 242 231 221 209 193 181 177 174 172 174 179 179 176 172 168 171 176 177 182 182 181 178 172 171]
[255 252 250 248 247 241 229 211 200 184 169 127 94 77 80 91 126 162 162 172 173 170 166 168 173 174 178 180 181 179 176 182]
[254 243 238 214 203 193 163 140 133 155 181 117 50 31 27 35 54 101 141 157 189 199 197 194 186 173 164 161 164 167 179 195]
[254 208 192 209 192 179 157 119 103 140 182 174 76 38 27 27 32 39 84 145 177 199 209 215 196 170 156 156 162 167 175 182]
[244 198 215 213 173 201 185 127 93 105 145 176 119 43 28 25 27 29 35 94 146 115 158 185 178 180 184 190 194 194 191 188]
[230 187 188 159 147 190 189 143 93 73 99 145 148 54 30 28 25 28 24 29 89 111 180 211 195 192 191 186 182 182 183 185]
[211 181 148 137 139 161 176 150 109 67 64 99 144 100 32 31 25 27 48 61 99 160 217 222 208 188 175 174 176 184 194 206]
[192 159 148 157 157 149 146 145 121 73 56 64 99 105 43 38 45 74 118 153 169 163 176 195 202 203 194 190 205 214 221 229]
[222 135 124 152 163 153 127 112 118 80 60 55 64 80 88 100 128 156 145 116 100 107 135 162 168 175 184 176 215 237 233 233]
[207 158 143 144 140 129 107 82 87 85 90 99 109 112 115 121 131 125 105 75 89 105 115 119 114 114 129 140 195 237 233 234]
[181 145 145 157 145 133 114 91 85 92 109 121 136 134 108 82 77 73 71 71 78 79 76 78 76 70 75 99 171 219 220 223]
[180 184 91 114 131 144 144 142 134 116 118 101 107 136 147 120 88 73 70 70 71 74 76 73 68 68 67 70 136 201 204 210]
[193 90 59 72 83 104 131 153 150 139 185 85 94 141 141 104 86 73 69 69 70 72 73 69 64 61 55 105 165 187 213]
[215 111 72 69 65 65 78 100 119 139 137 110 87 79 105 163 141 83 80 82 70 70 71 68 64 63 42 37 82 123 177 227]
[227 149 108 68 68 65 63 66 69 88 117 95 97 104 82 141 174 157 111 114 111 71 68 59 36 62 56 48 73 119 179 203]
[219 176 138 80 79 79 76 71 66 66 68 38 62 111 100 104 199 227 183 125 151 124 70 55 34 53 67 63 70 108 143 182]
[218 194 153 86 85 86 84 82 78 75 49 29 39 70 96 93 147 227 202 152 132 147 93 67 62 64 66 66 70 73 92 182]
[225 208 161 84 79 83 82 80 79 82 49 45 75 73 79 87 95 182 189 124 95 105 107 74 69 67 67 65 56 44 92 197]
[225 208 156 80 69 78 81 79 78 79 48 79 116 98 78 82 84 128 143 84 75 73 82 72 67 65 61 48 32 40 121 211]
[221 209 133 49 51 69 80 81 77 77 53 110 102 99 72 91 80 87 104 79 70 67 65 66 60 48 37 26 23 46 143 221]
[224 219 169 99 92 83 71 86 86 78 73 63 137 138 72 100 83 75 96 77 64 58 63 58 34 27 24 26 33 62 154 213]
[230 226 220 206 193 168 123 88 101 96 70 130 150 154 117 76 87 89 82 64 39 35 64 48 22 24 31 42 56 113 180 206]
[235 233 233 238 236 226 208 164 113 103 76 115 134 151 83 66 76 67 36 52 23 36 72 34 29 41 54 62 91 165 200 214]
[238 237 237 238 236 232 227 223 193 125 59 128 150 155 68 75 73 53 47 64 37 57 69 45 57 63 70 98 154 197 211 220]
[243 240 240 241 238 233 230 227 203 118 121 158 116 108 84 79 69 80 86 69 69 65 66 74 93 124 164 197 210 215 222]
[247 243 243 244 245 243 241 238 234 233 197 116 156 139 136 73 79 82 102 96 76 77 85 104 133 163 187 203 210 215 217 222]
[251 247 247 248 249 248 245 244 244 241 239 163 116 152 90 43 70 85 109 113 112 131 156 179 197 207 211 213 215 217 217 220]
[253 250 250 250 250 248 247 247 247 246 222 140 181 74 92 130 150 165 174 184 196 207 217 218 217 214 214 214 215 214 216]
[255 253 253 253 252 251 251 250 250 249 247 228 199 191 206 218 222 229 230 226 223 222 226 227 223 220 218 215 215 215 217]
[253 250 249 249 247 247 247 246 246 248 249 248 246 242 241 241 239 232 225 224 229 230 225 221 220 216 215 214 214 217]]
```



Curse of dimensionality

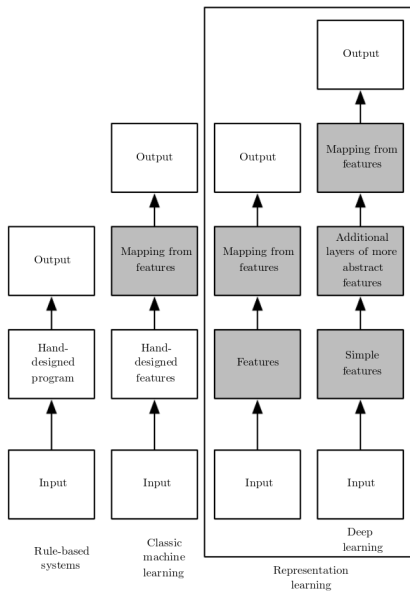
Typically, for a 256x256 color image:

$$3 \times 256 \times 256 \approx 2 \cdot 10^5 \text{ dimensions.}$$

In high dimensional spaces, the notion of locality is totally different.

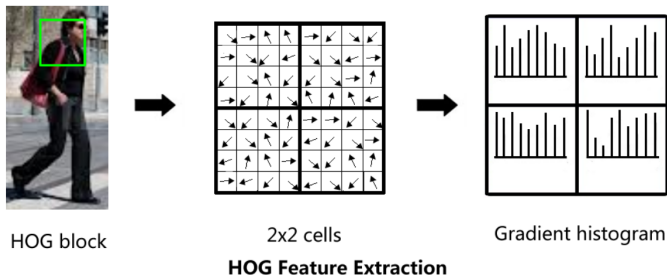
We need to learn an adapted representation.

Machine learning pipeline evolution



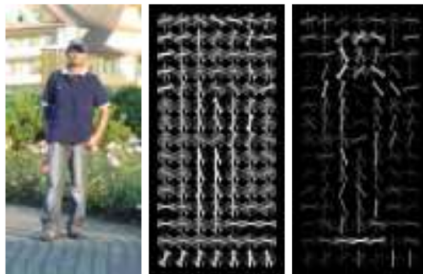
Computer vision classical pipelines

- ▶ Low-level representation approaches (HoG, SIFT, ...)
 - ▶ Hand-crafted features
 - ▶ supervised learning of classifier



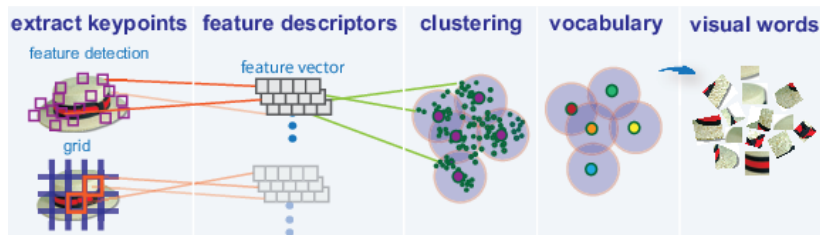
Computer vision classical pipelines

- ▶ Low-level representation approaches (HoG, SIFT, ...)
 - ▶ Hand-crafted features
 - ▶ supervised learning of classifier



Computer vision classical pipelines

- ▶ Low-level representation approaches (HoG, SIFT, ...)
 - ▶ Hand-crafted features
 - ▶ supervised learning of classifier
- ▶ Mid-level representation approaches (BoVW, FV, ...)
 - ▶ dense extraction of local visual descriptors
 - ▶ unsupervised learning of global representation
 - ▶ supervised learning of classifier



Computer vision classical pipelines

- ▶ Low-level representation approaches (HoG, SIFT, ...)
 - ▶ Hand-crafted features
 - ▶ supervised learning of classifier
- ▶ Mid-level representation approaches (BoVW, FV, ...)
 - ▶ dense extraction of local visual descriptors
 - ▶ unsupervised learning of global representation
 - ▶ supervised learning of classifier



Computer vision classical pipelines

- ▶ Low-level representation approaches (HoG, SIFT, ...)
 - ▶ Hand-crafted features
 - ▶ supervised learning of classifier
- ▶ Mid-level representation approaches (BoVW, FV, ...)
 - ▶ dense extraction of local visual descriptors
 - ▶ unsupervised learning of global representation
 - ▶ supervised learning of classifier
- ▶ End-to-end learning of representation approaches
 - ▶ end-to-end supervised learning: Deep Learning

Impact of deep learning on image recognition tasks

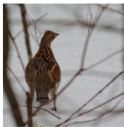
ILSVRC



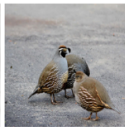
flamingo



cock



ruffed grouse



quail



partridge

...



Egyptian cat



Persian cat



Siamese cat



tabby



lynx

...



dalmatian



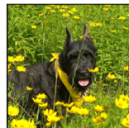
keeshond



miniature schnauzer



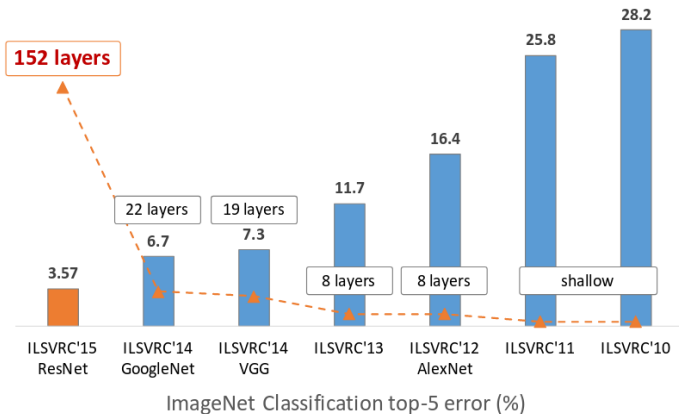
standard schnauzer



giant schnauzer

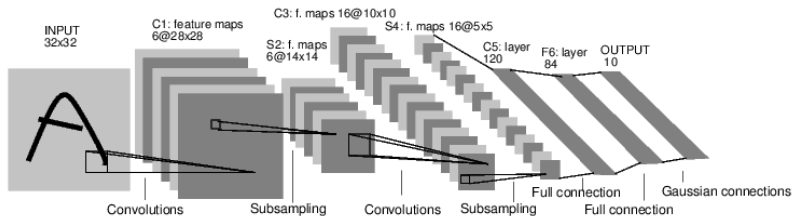
...

Impact of deep learning on image recognition tasks

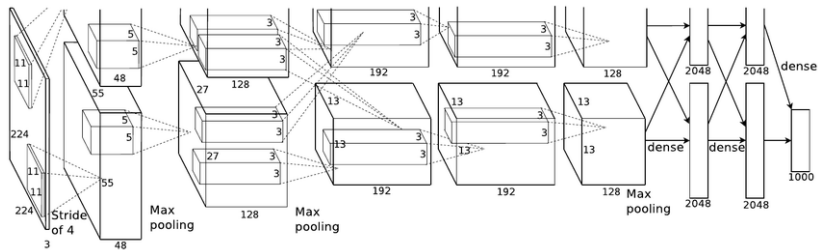


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

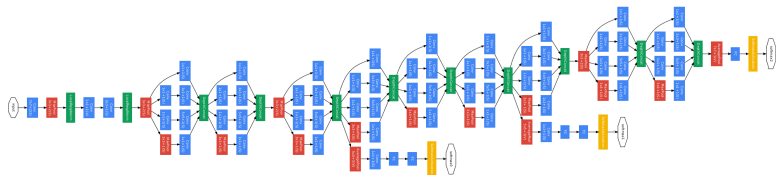
ConvNets (CNNs): LeNet (LeCun et al. 1998)



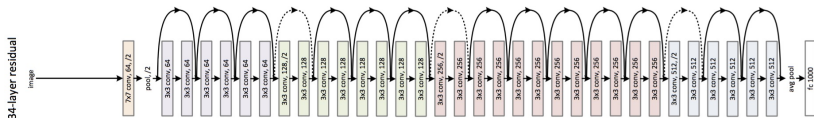
From CNNs to DNNs: AlexNet (Krizhevsky et al. 2012)



From CNNs to DNNs: GoogLeNet / Inception (Szegedy et al. 2015)



From CNNs to DNNs: ResNet (He et al. 2016)



(small version, biggest one: over 1000 layers)

Transfer Learning

When not enough training data is available, instead of training from scratch a neural network, use weights from models learned on other datasets (e.g. ImageNet).

There are usually two strategies:

- ▶ *finetuning the network*: use pretrained model as initialization rather than starting from random weights
- ▶ *fixed feature extractor*: use the neural network solely as a feature extractor, i.e. freeze all weights and replace the last linear layer by a layer learned on the dataset

Outline

Présentation du problème

(Très) courte introduction à la reconnaissance d'images

Détails du projet

Étapes principales

1. formattage des données
2. analyse globale, vue d'ensemble des données
3. formalisation du problème
4. solutions techniques
5. aller plus loin

Étapes principales

1. formattage des données
 - ▶ les données ne sont pas toujours très propres...
 - ▶ il faut les “standardiser”, par ex. convertir les XMLs en un unique CSV
2. analyse globale, vue d’ensemble des données
3. formalisation du problème
4. solutions techniques
5. aller plus loin

Étapes principales

1. formattage des données
2. analyse globale, vue d'ensemble des données
 - ▶ à quoi ressemblent les images ?
 - ▶ quelles sont les caractéristiques du jeu de données ?
3. formalisation du problème
4. solutions techniques
5. aller plus loin

Étapes principales

1. formattage des données
2. analyse globale, vue d'ensemble des données
3. formalisation du problème
 - ▶ quelle(s) métrique(s) choisir et pourquoi ?
 - ▶ quel protocole expérimental pour mesurer la performance des modèles ?
4. solutions techniques
5. aller plus loin

Étapes principales

1. formattage des données
2. analyse globale, vue d'ensemble des données
3. formalisation du problème
4. solutions techniques
 - ▶ comparaison des différents types de méthodes:
 - 4.1 classifieur linear : SVM, régression logistique, etc.
 - 4.2 descripteurs “hand-crafted” : HoG, SIFT, etc.
 - 4.3 representation intermédiaire : BoVW, Fisher Vectors, etc.
 - 4.4 réseaux neuronaux : MLP, CNN, etc.
 - ▶ quelles conclusions en tirer ?
5. aller plus loin

Étapes principales

1. formattage des données
2. analyse globale, vue d'ensemble des données
3. formalisation du problème
4. solutions techniques
5. aller plus loin
 - ▶ utilisation de jeux de données plus larges
 - ▶ exploitation des metadonnées
 - ▶ visualisation des features apprises par les classifieurs
 - ▶ etc.

Outils

1. Python
2. scikit-learn
3. PyTorch
4. OpenCV
5. autres...

Ressources

1. Moodle du cours et instructions du projet
2. User Guide de scikit-learn :
`https://scikit-learn.org/stable/user_guide.html`
3. Doc de PyTorch : `https://pytorch.org/docs/`
4. Tutoriels de PyTorch : `https://pytorch.org/tutorials/`

Questions ?

References I



He, Kaiming et al. (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.



LeCun, Yann et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.



Szegedy, Christian et al. (2015). “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.



Taigman, Yaniv et al. (2014). “Deepface: Closing the gap to human-level performance in face verification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708.