

# Towards landmarks prediction with Deep Network

Van-Linh LE<sup>1,3</sup>, Marie BEURTON-AIMAR<sup>1</sup>,  
Akka ZEMMARI<sup>1</sup>, Nicolas PARISEY<sup>2</sup>

[linhlv@dlu.edu.vn](mailto:linhlv@dlu.edu.vn), [van-linh.le@labri.fr](mailto:van-linh.le@labri.fr), [beurton@labri.fr](mailto:beurton@labri.fr)  
[akka.zemmari@labri.fr](mailto:akka.zemmari@labri.fr), [nicolas.parisey@inra.fr](mailto:nicolas.parisey@inra.fr)

<sup>1</sup>LaBRI-CNRS 5800, Bordeaux University, France

<sup>2</sup>IGEPP, INRA 1349 Rennes, France

<sup>3</sup>ITDLU, Dalat University, Vietnam

**ICPRS Conference**

Valparaíso, 22-24 May, 2018

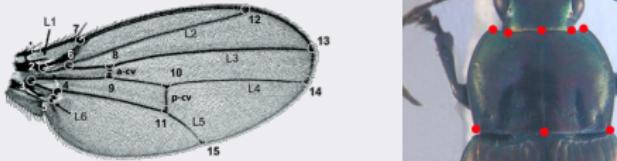


## Morphometry analysis

- ▶ Used to study the complex interaction between the evolution of insect and environmental factors.
- ▶ Characterize the common information of biological shape, such as, shape, sizes, or **landmarks**,....

## Landmark

- ▶ A kind of **point of interest**
- ▶ A specific point defined by biologist. For example, intersection of veins on fly wing, the corner of beetle's pronotum,....



# Dataset



- ▶ Images have been taken from 293 **beetles**, separate into 5 parts (images),
- ▶ Format: 2D in RGB color,
- ▶ Focus on **pronotum** images.



(a) Left mandible



(b) Right mandible



(c) Body

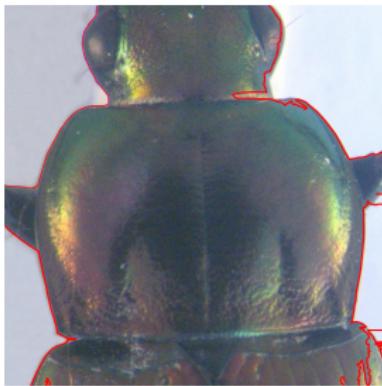


(d) Head

# Problems



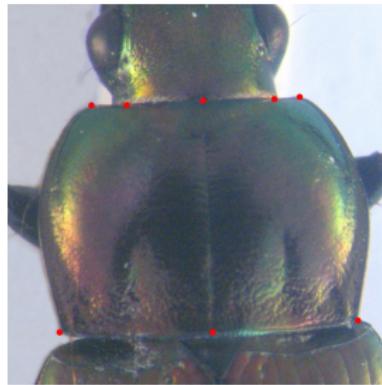
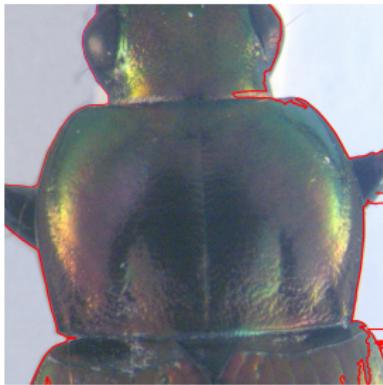
With un-segmentable images:



# Problems



With un-segmentable images:



**How to predict the landmarks coordinates?**

# Content



## Deep learning and Convolutional Neural Networks

Deep learning

Convolutional neural networks (CNNs)

## Proposed method

Network architectures

Data augmentation

Training from scratch

## Result

Fine-tuning

## Conclusion



## Definition<sup>1</sup>

- ▶ A class of machine learning,
- ▶ Use a cascade of multiple layers for feature extraction and transformation,
- ▶ Learn multiple levels of representation in supervised or unsupervised.

<sup>1</sup> Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015



## Definition<sup>1</sup>

- ▶ A class of machine learning,
- ▶ Use a cascade of multiple layers for feature extraction and transformation,
- ▶ Learn multiple levels of representation in supervised or unsupervised.

## Applications<sup>1</sup>

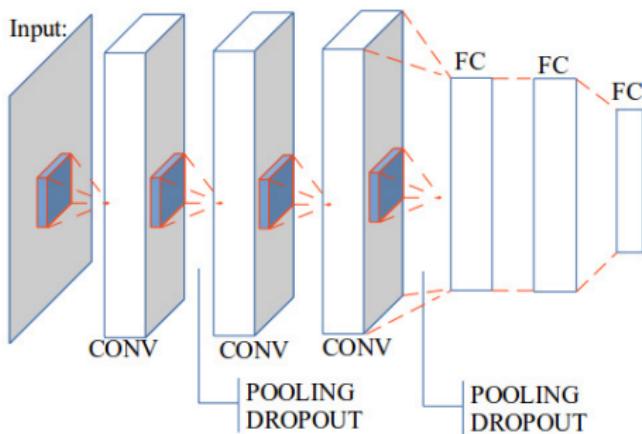
- ▶ Computer vision (image recognition and classification)
- ▶ Speech recognition
- ▶ Question answering, language translation

<sup>1</sup> Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015

# CNNs



- ▶ Consists an input, an output and multiple hidden layers<sup>1</sup>
- ▶ Arranges the data in 3 dimensions: *width, height and depth*
- ▶ Classical layers: convolutional layers (**CONV**), pooling layers (**POOLING**), dropout layers (**DROPOUT**), full-connected layers (**FC**), ...



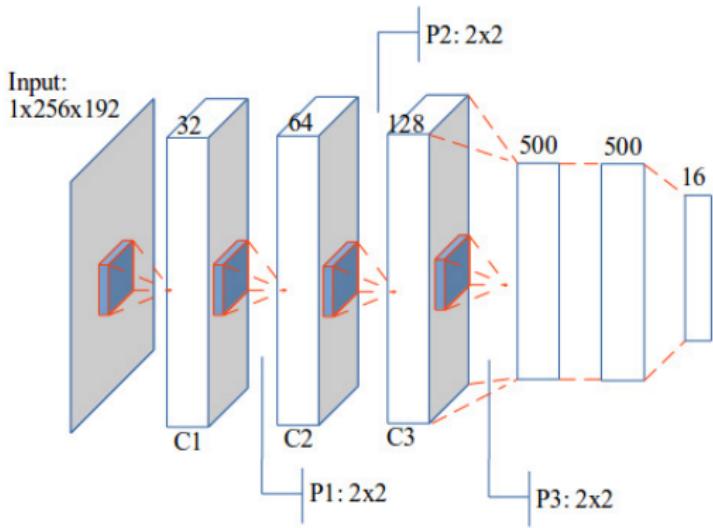
<sup>0</sup> Y. LeCun et al, "Convolutional networks and applications in vision", 2010.

# Network architecture



Elementary block:

- ▶ A CONV layer,
- ▶ A maximum POOLING layer,
- ▶ A DROPOUT layer

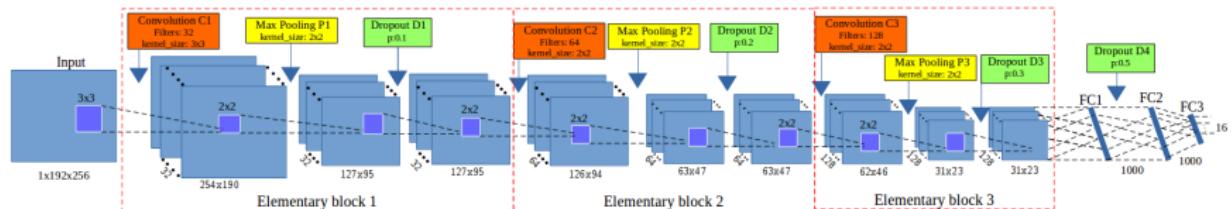


# Network architecture



The proposed model:

- ▶ **Three** elementary blocks,
- ▶ **Three** full-connected (FC) layers
- ▶ A dropout layer was inserted between the first of two FCs



# Data augmentation



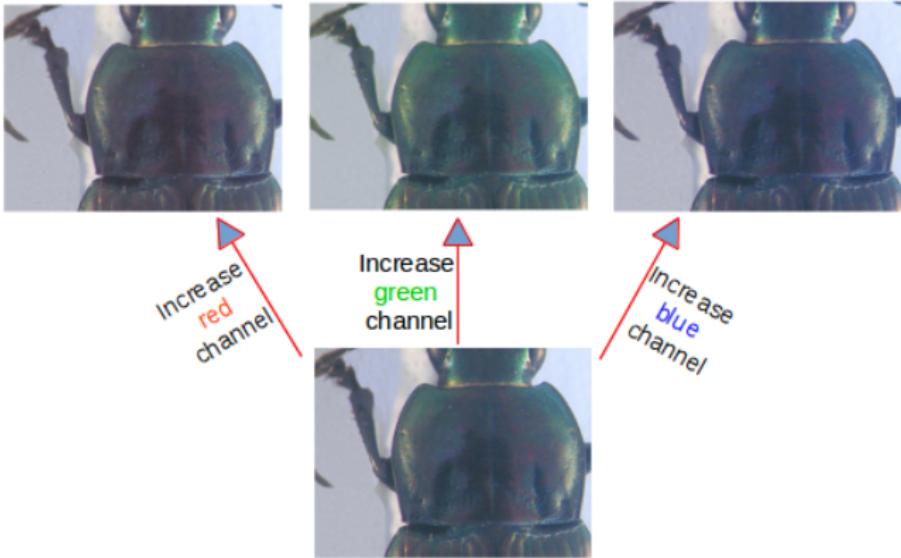
Dataset: 293 pronotum images in RGB format.

# Data augmentation



## Augmentation methods:

- ▶ Increase the value of each channel,

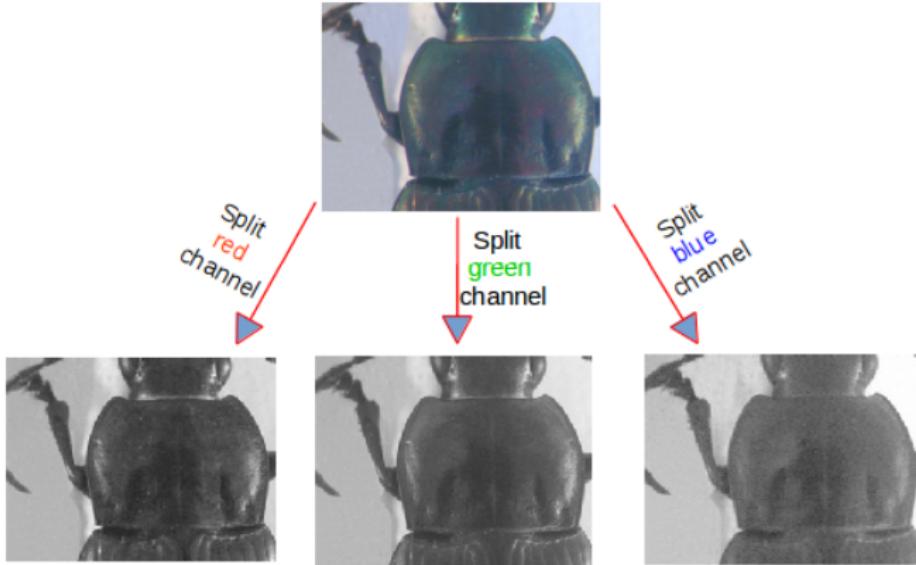


# Data augmentation



Augmentation methods:

- ▶ Split the channels.



# Data augmentation

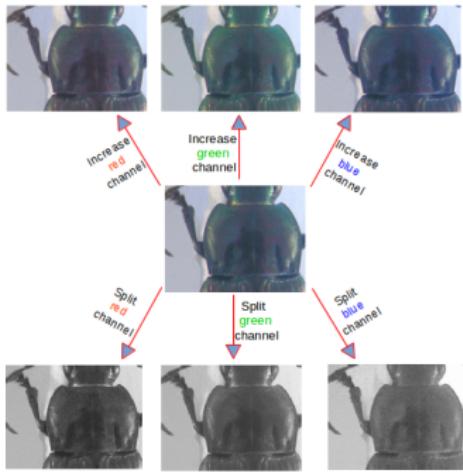


Dataset: 293 pronotum images in RGB format.

Augmentation methods:

- ▶ Increase the value of each channel,
- ▶ Split the channels.

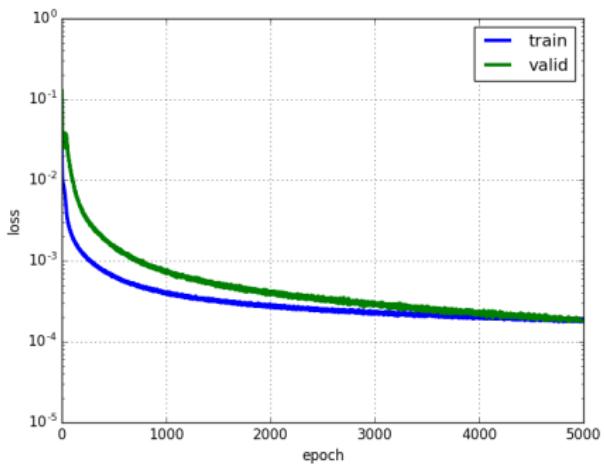
Total:  $293 \times 7 = 2,051$  images



# Training



- ▶ Model: the third model in 5,000 epochs<sup>1</sup>
- ▶ Training dataset: 1,820 images ( $260 \times 7$ )
- ▶ Images shows training and validation losses of the models.  
Blue curves are training losses, green curves are validation losses.
- ▶ Training time: 3 hours using NVIDIA TITAN X card.



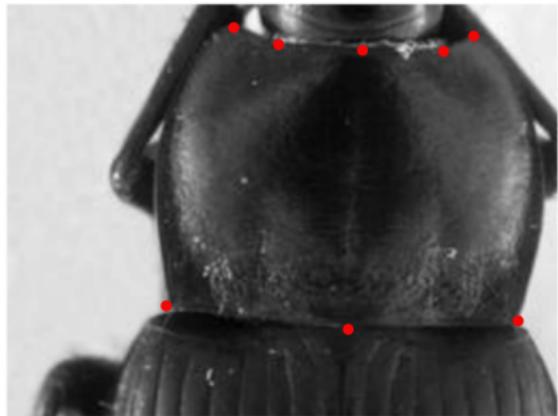
<sup>1</sup>An epoch is a single pass through the full training set.

# Result

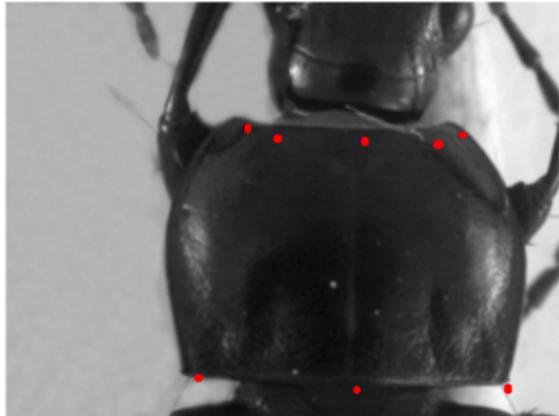
## Landmarks on images



Images show the result on testing images.



(a)



(b)

# Result

Average distance



- ▶ Run the trained model to predict the landmarks on testing images,
- ▶ Calculate the distance between predicted landmarks and corresponding manual landmarks,
- ▶ Compute the average distance of all images per landmark.

#Landmark	Distance (in pixels)
1	4.002
2	4.4831
3	4.2959
4	4.3865
5	4.2925
6	5.3631
7	4.636
8	4.9363

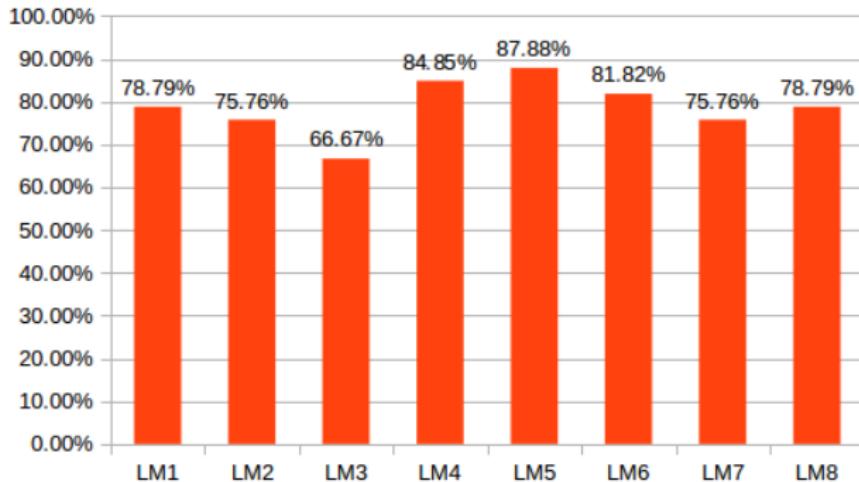
# Result

Statistic on acceptable predicted landmarks



Chart shows the proportion of acceptable predicted landmarks

- ▶ Average accuracy: ~ 75%
- ▶ Highest accuracy: 87.88%
- ▶ Lowest accuracy: 66.67%

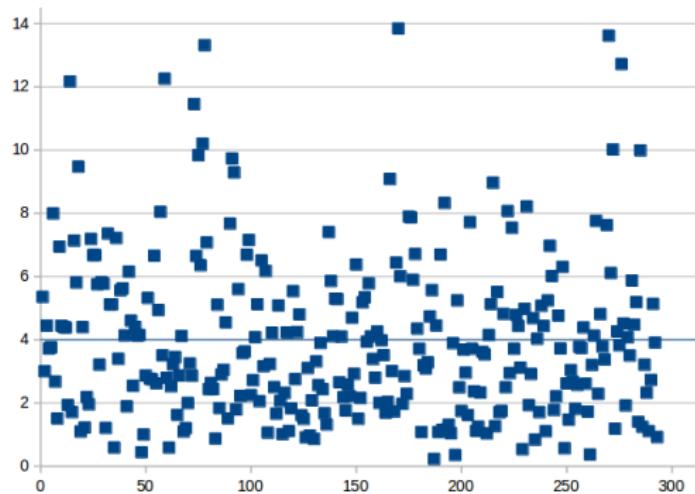


# Result

Distribution of distance on the first landmark



- ▶ Good prediction: 56.66%
- ▶ Acceptable prediction: 40.27%
- ▶ Bad prediction: 3.07%



# Result

Comparing with related works



Quality metrics: coefficient of determination ( $r^2$ ), explained variance (EV), Pearson correlation.

Metric	$r^2$	EV	Pearson
Cintas et al. <sup>2</sup>	0.884	0.951	0.976
Proposed architecture	<b>0.9952</b>	<b>0.9951</b>	<b>0.9974</b>

<sup>2</sup> Cintas, "Automatic ear detection and feature extraction using geometric morphometrics and convolutional neural networks," IET Biometrics, vol. 6, no. 3, pp. 211–223, 2016

# Conclusion



## Conclusion

- ▶ Proposed a CNN to predict the landmarks on pronotum images.
- ▶ Proposed procedure to augment the dataset.
- ▶ The location of the predicted landmarks are acceptable with high accuracy ( $\sim 75\%$ ). It allows to replace manual landmarks.

# Conclusion



## Conclusion

- ▶ Proposed a CNN to predict the landmarks on pronotum images.
- ▶ Proposed procedure to augment the dataset.
- ▶ The location of the predicted landmarks are acceptable with high accuracy ( $\sim 75\%$ ). It allows to replace manual landmarks.

## Future works

Continue improving the landmarks coordinates by continuing on deep learning, *for example*, using transfer learning.



Thank you for attention!