

Lesson 4.6: Application: Image Classification

Class 1 Time series forecasting

Class 2 Data stream mining
in Weka and MOA

Class 3 Interfacing to R and other data
mining packages

Class 4 Distributed processing with
Apache Spark

Class 5 Scripting Weka in Python

Lesson 4.1 What is distributed Weka?

Lesson 4.2 Installing with Apache Spark

Lesson 4.3 Knowledge Flow templates I

Lesson 4.4 Knowledge Flow templates II

Lesson 4.5 Moving to a cluster

Lesson 4.6 Application: Image
Classification

Lesson 4.6: Application: Image Classification

- ❖ **Image features** are a way of describing an image using numbers
- ❖ For example:
 - *How bright is the image (f_1)?*
 - *How much yellow is in the image (f_2)?*
 - *How much green is in the image (f_3)?*
 - *How symmetrical is the image (f_4)?*



f_1	50%
f_2	50%
f_3	10%
f_4	100%



f_1	50%
f_2	2%
f_3	65%
f_4	50%

Lesson 4.6: Application: Image Classification

- ❖ **Image filters** extract the same features for a set of images

```
@relation butterfly_vs_owl
@attribute filename string
@attribute class {BUTTERFLY,OWL}
@data
mno001.jpg,BUTTERFLY
mno002.jpg,BUTTERFLY
mno003.jpg,BUTTERFLY
mno004.jpg,BUTTERFLY
owl1001.jpg,OWL
owl1002.jpg,OWL
owl1003.jpg,OWL
owl1004.jpg,OWL
```

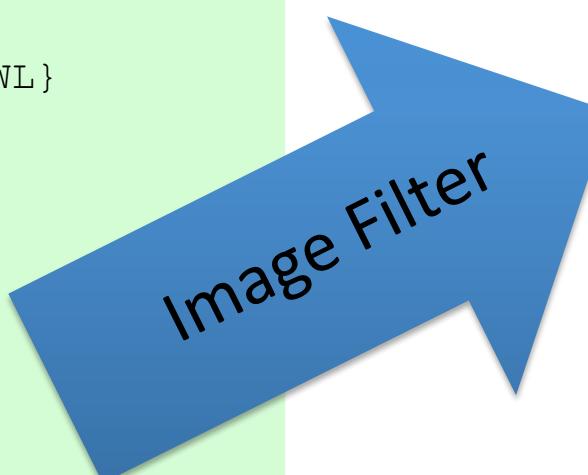


Image Filter

```
@relation butterfly_vs_owl
@attribute filename string
@attribute f1 numeric
@attribute f2 numeric
@attribute f3 numeric
@attribute class {BUTTERFLY,OWL}

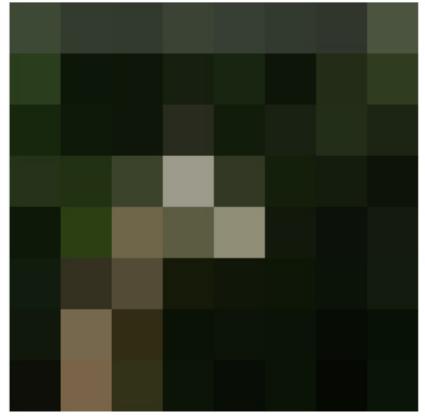
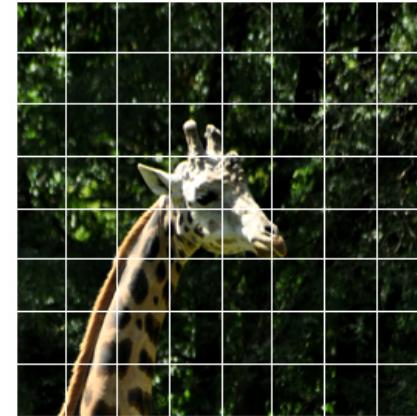
@data
mno001.jpg,3,7,0,BUTTERFLY
mno002.jpg,1,2,0,BUTTERFLY
mno003.jpg,3,4,0,BUTTERFLY
mno004.jpg,6,3,0,BUTTERFLY
owl1001.jpg,3,5,0,OWL
owl1002.jpg,7,3,0,OWL
owl1003.jpg,3,5,0,OWL
owl1004.jpg,7,5,1,OWL
```

Lesson 4.6: Application: Image Classification

1. Install **imageFilters** package using the Package Manager
2. Place all of your images in the same directory
3. Create an ARFF file with the first string attribute being an image's filename and each instance being an image and its class
4. Open the ARFF file in the WEKA Explorer
5. Select an image filter from **filters/unsupervised/instance/imagefilters**
6. Set the filter's **imageDirectory** option to the correct directory
7. Click the Apply button
8. Repeat 5-7 if you wish to apply more than one filter
9. (Optional) Remove the first filename attribute
10. Select a classifier and perform some experiments

Lesson 4.6: Application: Image Classification

- ❖ Different image filters extract different measurements
- ❖ Example: **ColorLayoutFilter**
 - *Image is divided into 64 blocks*
 - *Average colour is assigned to each block*
 - *Features are computed from the average colours*
- ❖ Example: **EdgeHistogramFilter**
 - *Edges are “lines” or “discontinuities” in image*
 - *Features measure the direction of the edges*



Lesson 4.6: Application: Image Classification

❖ Summary

- *Image features are mathematical properties of images*
- *Image filters can be applied to calculate image features for an entire dataset of images*
- *Different features measure different properties of the image*
- *Experimenting with WEKA can help you identify the best combination of image feature and classifier for your data*

❖ References

- *Mathias L., Chatzichristofis S. A. Lire: Lucene Image Retrieval – An Extensible Java CBIR Library. In Proc. of the 16th ACM International Conference on Multimedia, pp. 1085-1088, Vancouver, Canada, 2008*
- *Manjunath B., Ohm J.R., Vasudevan V.V., Yamada A. Color and texture descriptors. IEEE trans. on Circuits and Systems for Video Technology 11, 703–715, 2001*