A MARFCLEF Approach to LifeCLEF 2015 Tasks

Serguei A. Mokhov

Concordia University, Montreal, Canada,
mokhov@cse.concordia.ca

Abstract. We make the first use of MARF of fast signal-processing and related techniques for LifeCLEF 2015 identification tasks. We build an application based on a pattern recognition pipeline implemented in an open-source Modular A* Recognition Framework (MARF). MARF is also the name of the team in this submission. For that purpose to test and select among available algorithm a set of suitable algorithms. This is the first implementation of the application we call `MARFCLEFApp` tested on a very small subset of algorithms available. The approach covers Bird-, Plant-, and FishCLEF tasks. It was expected the bird task would be the best for the presented approach given MARF's original intent for audio recognition. However, lack of enough run-time it turned out to be the worst one and is under the investigation. Processing FishCLEF however yield the best of the three tasks, which was expected to be the worst. Team MARF's results for FishCLEF were the 2nd team after with the Run 1 being the best of the three.

1 Introduction

1.1 Introduction to MARF

Modular Audio Recognition Framework (MARF) is an open-source collection of pattern recognition APIs and their implementation for unsupervised and supervised machine learning and classification written in Java [6]. One of its design purposes is to act as a testbed to try out common and novel algorithms found in literature and industry for sample loading, preprocessing, feature extraction, and training and classification tasks. One of the main goals and design approaches of MARF is to provide scientists with a tool for comparison of the algorithms in a homogeneous environment and allowing the dynamic module selection (from the implemented modules) based on the configuration options supplied by applications. Over the course of several years MARF accumulated a fair number of implementations for each of the pipeline stages allowing reasonably comprehensive comparative studies of the algorithms *combinations*, and studying their *combined* behavior and other properties when used for various pattern recognition tasks. MARF is also designed to be very configurable while keeping the generality and some sane default settings to "run-off-the-shelf" well. MARF and its derivatives, and applications were also used beyond audio processing tasks due to the generality of the design and implementation in [5,7] and other works.

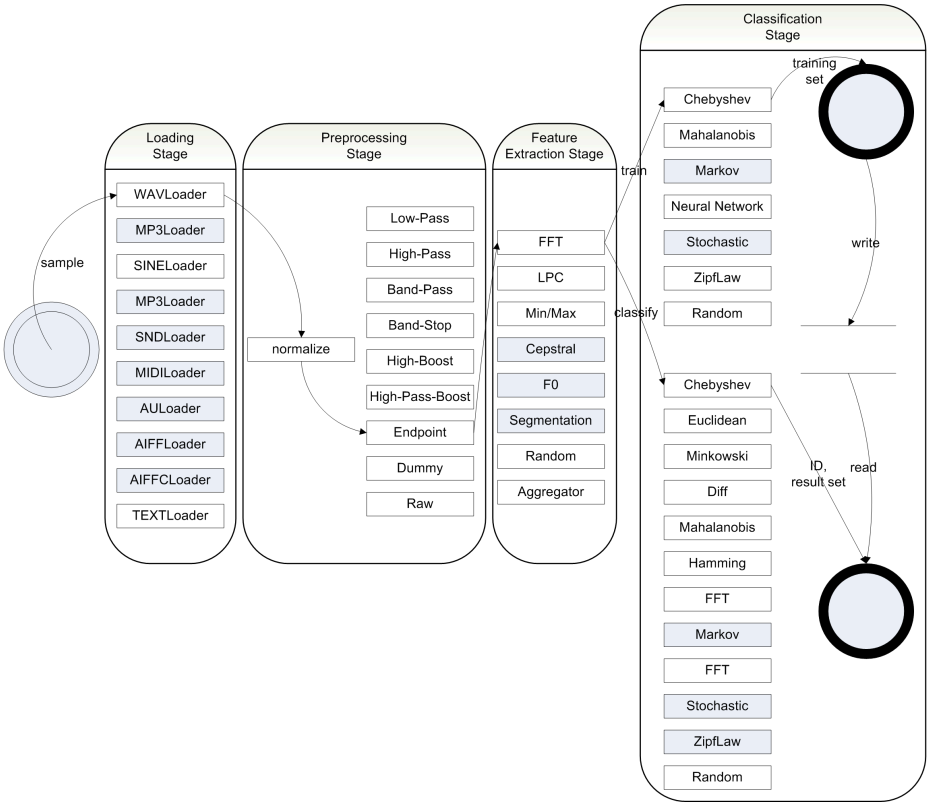

Fig. 1. MARF's Pattern Recognition Pipeline

The methodology behind `MARFCLEFApp` builds on the successes and failures of the previous similar applications used for different tasks, such as MARFCAT [9], `HEp2IdentApp` and `MARFIIFApp` and others were the source of inspiration for all three tasks. MARF [16] is the core framework behind them all.

1.2 Common Index File Format

All tasks' data are annotated by a unified XML index format, so technically any task's metadata can be encoded in it. The input index format is otherwise inherited from MARFCAT [15,13]. This format looks like the below.

```
<?xml version="1.0" encoding="UTF-8"?>
<dataset generated-by="./collect-files-meta.pl" generated-on="2011-11-13T01:23:36">
        <description>
                <file-type-tool>
                </file-type-tool>
                <find-tool>
                </find-tool>
                <marf-tool>
                </marf-tool>
        </description>
        <file id="3" path="bird/train/LIFECLEF2015_BIRDAMAZON_XC_WAV_RN410.wav">
                <meta>
                        <type>Bourne shell script text executable</type>
                        <length lines="" words="" bytes="861" />
                </meta>
                <location line="" fraglines="">
                        <meta>
<Audio>
  <MediaId>410</MediaId>
  <FileName>LIFECLEF2015_BIRDAMAZON_XC_WAV_RN410.wav</FileName>
  <ClassId>arlrfh</ClassId>
  <Date>2000-04-21</Date>
  <Time>am</Time>
  <Locality>Finca La Pradera, Parroquia Remolinos, Municipio Puerto Lopez, Meta</Locality>
  <Latitude>NULL</Latitude>
  <Longitude>NULL</Longitude>
  <Elevation>400</Elevation>
  <Author>Bernabe Lopez-Lanus</Author>
  <AuthorID>BYLLSJRVDZ</AuthorID>
  <Content>Call</Content>
  <Comments />
  <Quality>3</Quality>
  <Year>BirdCLEF2015</Year>
  <BackgroundSpecies />
  <Order>Cuculiformes</Order>
  <Family>Cuculidae</Family>
  <Genus>Crotophaga</Genus>
  <Species>major</Species>
  <Sub-species />
  <VernacularNames>Greater Ani</VernacularNames>
</Audio>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
                        </explanation>
                </location>
        </file>
        <file id="5" path="bird/test/LIFECLEF2014_BIRDAMAZON_XC_WAV_RN3165.wav">
                <meta>
                        <type>WAV</type>
                        <length lines="" words="" bytes="13620" />
                </meta>
                <location line="" fraglines="">
                        <meta>
<Audio>
  <MediaId>3165</MediaId>
  <FileName>LIFECLEF2014_BIRDAMAZON_XC_WAV_RN3165.wav</FileName>
  <Date>2007-10-28</Date>
  <Time>13:00</Time>
  <Locality>Taquaril - Petrpolis - RJ</Locality>
  <Latitude>-22.3023</Latitude>
  <Longitude>-43.077</Longitude>
  <Elevation>1100</Elevation>
  <Author>Ricardo Gagliardi</Author>
  <AuthorID>YRDADNREOP</AuthorID>
  <Year>BirdCLEF2014</Year>
</Audio>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
                        </explanation>
                </location>
        </file>
        <file id="6" path="plant/train/23546.png">
                <meta>
```

```
                        <type>ASCII English text</type>
                        <length lines="6964" words="24690" bytes="199984" />
                </meta>
                <location line="" fraglines="">
                        <meta>
<Image>
  <ObservationId>13075</ObservationId>
  <MediaId>23546</MediaId>
  <Vote>3</Vote>
  <Content>Flower</Content>
  <ClassId>4742</ClassId>
  <Family>Rosaceae</Family>
  <Genus>Filipendula</Genus>
  <Species>Filipendula vulgaris Moench</Species>
  <Author>gilles carcasses</Author>
  <Date>2013-7-16</Date>
  <Location>Ipiais-Rhus</Location>
  <Latitude>49.12491</Latitude>
  <Longitude>2.06075</Longitude>
  <YearInCLEF>PlantCLEF2015</YearInCLEF>
  <ObservationId2014 />
  <ImageId2014 />
  <LearnTag>Train</LearnTag>
</Image>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
                        </explanation>
                </location>
        </file>
        <file id="77779" path="fish/train/01465f8f61db58564cd37ce2dfc519c5#201106090830_frame_966.png">
                <meta>
                        <type>PNG</type>
                        <length lines="" words="" bytes="7449" />
                </meta>
                <location line="" fraglines="">
                        <meta>
<video id="01465f8f61db58564cd37ce2dfc519c5#201106090830" location="NPP3" camera="1">
   <frame id="966">
      <object fish_species="Dascyllus Aruanus" h="71" w="58" x="5" y="171"/>
   </frame>
</video>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
   <!--frame id="966">
      <object fish_species="Dascyllus Aruanus" h="71" w="58" x="5" y="171"/>
      <object fish_species="Chaetodon Lununatus" h="49" w="65" x="511" y="326"/>
      <object fish_species="Dascyllus Reticulatus" h="48" w="31" x="591" y="123"/>
   </frame-->
                        </explanation>
                </location>
                <location line="" fraglines="">
                        <meta>
<video id="01465f8f61db58564cd37ce2dfc519c5#201106090830" location="NPP3" camera="1">
   <frame id="966">
      <object fish_species="Chaetodon Lununatus" h="49" w="65" x="511" y="326"/>
   </frame>
</video>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
                        </explanation>
                </location>
                <location line="" fraglines="">
```

```
                        <meta>
<video id="01465f8f61db58564cd37ce2dfc519c5#201106090830" location="NPP3" camera="1">
   <frame id="966">
      <object fish_species="Dascyllus Reticulatus" h="48" w="31" x="591" y="123"/>
   </frame>
</video>
                        </meta>
                        <fragment>
                        </fragment>
                        <explanation>
                        </explanation>
                </location>
        </file>
```

1.3 Generalities

All MARF-based runs of `MARFCLEFApp` are fully automatic. This includes the generation of index files as per earlier section, and preparing for training and classification. No manual pre- or post-processing are done. Only adaption of the MARFCAT and related apps to CLEF task and writing specialized loader plug-ins were done in preparation for the runs.

The algorithms There was not enough time or team human resources available to do a better search for the most suitable algorithm/parameter combination, so best options were picked from past runs in other applications.

The application is written in Java, and scripting is done using `tcsh`, `perl`, `bash`, and `gmake` Makefiles. Experiments ran on OS X 10.10.3 and Scientific Linux 6.x.

1.4 BirdCLEF

The runs for BirdCLEF [2] were primarily inspired by the related work on audio from the 2008 paper [6] that targeted spoken accent and gender identification as an extension from the text-independent speaker identification task, based on MARF from 2002–2008, with recent improvements.

Unfortunately, picking the configurations from the previous work on the large number of classes, did not work well in the BirdCLEF runs in 2015 as was hoped and there was no run-time available to comprehensively test all possible combinations available like was done in [6].

The meta-data approach inspired by the task description seemed an interesting idea and was also attempted. It has achieved 70% precision on the training data, but low precision results on the testing data. `MARFCLEFApp` uses the same mechanism to load the metadata as was done in MARFCAT for code analysis [10,12,13], which was a successful experiment. There the textual data are interpreted as a wave signal and processed by the rest of the pipeline like for normal audio signal.

The author suspects a combination of the overfitting problem as well as mean clustering for many classes such as 1000 species contribute to the low precision. There are more comprehensive experiments currently running to answer these questions and doubts.

We did 4 basic runs that were submitted:

Run 1 Classical audio-only, as in the sense of MARF and [6]. This run has produced the best precision of the four submissions in this work.
Run 2 MARFCAT-like meta-data only is the second run.
Run 3 This run combines meta-data and audio using a new merging new `MetaWAVLoader` that concatenates the signals from the meta-data and audio.
Run 4 This run is audio only with with quality categories taken into account.

There are other runs as well of course with the permutations of audio and metadata, algorithms, and categories.

Example statistics on meta-data-only (of the training set):

```
guess,run,config,good,bad,%,recall,mca
1st,1,--birds bird-train-xml.xml -wav -44kHz -nopreprep -raw -fft -cos,16808,7799,68.31,100.00,
1st,2,--birds bird-train-xml.xml -wav -44kHz -nopreprep -raw -fft -eucl,8061,16546,32.76,100.00,
2nd,1,--birds bird-train-xml.xml -wav -44kHz -nopreprep -raw -fft -cos,16814,7793,68.33,100.00,
2nd,2,--birds bird-train-xml.xml -wav -44kHz -nopreprep -raw -fft -eucl,8083,16524,32.85,100.00,
```

1.5 FishCLEF

MARF's use in FishCLEF [1] was surprisingly better than in the BirdCLEF's task.

As a part of this task, first the `.flv` files were translated to a series of `.png` images, one per frame, using `ffmpeg`[1]. The `.flv` videos (using `ffmpeg`) are saved as `.png` images with the settings quoted below. As a result, the video task in a way became an image task.

The specific command used was, for example:

```
ffmpeg \
    -i 01465f8f61db58564cd37ce2dfc519c5#201106090830_0.flv \
    -r 25 \
    -vcodec png \
    -pix_fmt rgb32 \
    01465f8f61db58564cd37ce2dfc519c5#201106090830_0_Frame_%d.png
```

The resulting `*_Frame_%d.png` were indexed in the common MARFCAT-IN XML format described in Section 1.2. A separate index was created for the provided training images per species.

The quick experiments conducted here were:

Run 1 Train on provided png images.
Run 2 Train on png image frames off the train videos.
Run 3 Train on provided png images and png image frames off the train videos.

[1] https://www.ffmpeg.org/

The above training runs were subsequently run on the 72 testing videos to produce the submitted run files.

For each frame that has multiple fish objects, we handle the meta-data similarly to dealing with multiple locations of CVEs in MARFCAT [10]. When a signal (spectrum) of each fish species in the result set is detected, it is output if its score is above certain threshold of the distance or similarity classifiers. This allows outputting multiple fish species detections per frame.

The results from Run 1 above provided the best precision and counting score on the testing data, per the analysis from the task organizers [1]. Normalized counting score per fish species [1] for Run 1 appear to be the best among the three MARF runs as well except species 12 and 15, in which Run 2 significantly outperforms.

As it happens the current submission the FishCLEF task yielded best results for the MARF team among the three tasks. The MARF team is also the second top team in this task. This was surprising at first given sub-par performance in BirdCLEF, but supports hypothesis that the default setup for MARF and `MARFCLEFApp` is not set properly for large number of classes: FishCLEF has only 15 species as opposed to 1000 in BirdCLEF.

One of the current limitations in the `MARFCLEFApp` implementation for this task is the lack of bounding box calculation (run output currently has hardcoded default values). This annotation is planned to be done in the future versions. This is a difficult problem in MARF-based application without image processing facilities. One approach would be to scan with a certain step from the top-left corner to find maximum probability signal for each detected fish species and remember the coordinate. Repeat the scan from the bottom-right corner. The two resulting maximum-signal coordinates would from the bounding box. This process is inherently parallelizable and can be decoupled from the main species detection task while computing bounding boxes in the background.

1.6 PlantCLEF

The PlantCLEF [3] uses the same parameters as FishCLEF, but its runs were submitted late and were not included into the overall analysis by the organizers as of this writing.

Given the focus on PlantCLEF was on observations rather than images, the meta-data on-load was internally adjusted to create finer classes based on media type.

Thus we subcategorize the media per species observations as follows (assuming a placeholder "species" *Rose* (represented as ClassID from the metadata)): `Rose_Flower`, `Rose_Stem`, `Rose_Scan`, `Rose_etc` internally, but `MARFCLEFApp` maps them all to *Rose* at classification. This was done to separate significant difference between media and species.

Since the run files require ranking of results up to 1000 species the above ≈ 6000 classes would be picked from the fused result set and the topmost likely species. The underscores and types are stripped outputting the final results.

Similarly to BirdCLEF, we add _Vote to the above for quality-based fingerprinting to separate samples with better quality from the worse, but don't discard them entirely. In this case an internal class representation becomes something like `Rose_Flower_5`. Unfortunately, per meta-data it appears that not all entries had `<Vote>` available. The same option `-qod` from the birds task works here as well.

Runs Internally runs are all flattened to be image-based as described by the subdivision of 1000 classes into media type and optionally vote-based sub-classes. They are reconstructed back into the observations-based output during the result-set fusion and raking.

Run 1 Plain image
Individual result for all tasks looks something like:

```
                 File: plant-train/110805.jpg
              Path ID: 9812
               Config: plant-train-jpg.xml plant-train-jpg.xml
                       -wav -44kHz -silence-noise -raw -fft -cos
      Processing time: 0d:0h:0m:0s:227ms:227ms
         Subject's ID: 275
   Subject identified: 5623
    Subject's species: Tilia cordata Mill.
ResultSet: [suppressed; enable debug mode to show]...
Expected subject's ID: 766 (possible: [766])
     Expected subject: 6425
     Expected species: Cypripedium calceolus L.
       Second Best ID: 276
     Second Best Name: 572
  Second Best species: Centaurea calcitrapa L.
            Date/time: Thu May 07 11:32:39 EDT 2015
```

Run 2 Meta-data only, similarly as per birds.
Run 3 Image+metadata
Run 4 Image+metadata+vote where available.

Each of the run also have a corresponding "flat" image run file with the underscore categories stripped out.

2 Conclusion

We would like to thank LifeCLEF task organizers [4] for putting together the challenge, data sets, and results evaluation. A companion arXiv paper is being prepared with the more complete results collection.

We use MARF as a library for signal process and applications built on top of it. The details of the algorithms used and their selection and ranking are exemplified in [6]. This approach is likewise similar in a way to where MARF was applied to file type analysis for forensic purposes [11] using machine learning and assuming each file is a sort of a signal on Unix systems as compared to the traditional `file` utility as well as to writer identification [14], natural language [8], and code analysis [12,13]. We treat all media: audio, imagery, and text as a waveform signal and apply appropriate techniques for noise and silence removal, frequency analysis and fast (and slow) classifiers for distances, similarity, and others.

FishCLEF task appears to be better handled than BirdCLEF and presumably PlantCLEF, requiring review of the setup for the large number of classes for better clustering and the experiments currently under way to confirm that.

Limitations Not all planned experiments are included to improve the precision, some of which are still running at the time of this writing. Public release of the results and the application nonetheless is planned to address overfitting and precision issues.

Acknowledgments

This work is partially funded by the Faculty of Engineering and Computer Science, Concordia University, Montreal, Canada. The author acknowledges the LifeCLEF organizers for their support.

References

1. Concetto, S., Fisher, B., Boom, B.: LifeCLEF Fish identification task 2015. In: CLEF working notes 2015 (2015)
2. Goëau, H., Glotin, H., Vellinga, W.P., Rauber, A.: LifeCLEF Bird identification task 2015. In: CLEF working notes 2015 (2015)
3. Goëau, H., Joly, A., Bonnet, P.: LifeCLEF Plant identification task 2015. In: CLEF working notes 2015 (2015)
4. Joly, A., Müller, H., Goëau, H., Glotin, H., Spampinato, C., Rauber, A., Bonnet, P., Vellinga, W.P., Fisher, B.: LifeCLEF 2015: multimedia life species identification challenges. In: Proceedings of CLEF 2015 (2015)
5. Mokhov, S.A.: On design and implementation of distributed modular audio recognition framework: Requirements and specification design document. [online] (Aug 2006), project report, `http://arxiv.org/abs/0905.2459`, last viewed April 2012
6. Mokhov, S.A.: Study of best algorithm combinations for speech processing tasks in machine learning using median vs. mean clusters in MARF. In: Desai, B.C. (ed.) Proceedings of C3S2E'08. pp. 29–43. ACM, Montreal, Quebec, Canada (May 2008)
7. Mokhov, S.A.: Towards security hardening of scientific distributed demand-driven and pipelined computing systems. In: Proceedings of the 7th International Symposium on Parallel and Distributed Computing (ISPDC'08). pp. 375–382. IEEE Computer Society (Jul 2008)
8. Mokhov, S.A.: L'approche MARF à DEFT 2010: A MARF approach to DEFT 2010. In: Proceedings of the 6th DEFT Workshop (DEFT'10). pp. 35–49. LIMSI / ATALA (Jul 2010), DEFT 2010 Workshop at TALN 2010; online at `http://deft.limsi.fr/actes/2010/pdf/2_clac.pdf`

9. Mokhov, S.A.: MARFCAT – MARF-based Code Analysis Tool. Published electronically within the MARF project, `http://sourceforge.net/projects/marf/files/Applications/MARFCAT/` (2010–2015), last viewed February 2014
10. Mokhov, S.A.: The use of machine learning with signal- and NLP processing of source code to fingerprint, detect, and classify vulnerabilities and weaknesses with MARFCAT. Tech. Rep. NIST SP 500-283, NIST (Oct 2011), report: `http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909407`, online e-print at `http://arxiv.org/abs/1010.2511`
11. Mokhov, S.A., Debbabi, M.: File type analysis using signal processing techniques and machine learning vs. `file` unix utility for forensic analysis. In: Goebel, O., Frings, S., Guenther, D., Nedon, J., Schadt, D. (eds.) Proceedings of the IT Incident Management and IT Forensics (IMF'08). pp. 73–85. LNI140, GI (Sep 2008)
12. Mokhov, S.A., Paquet, J., Debbabi, M.: The use of NLP techniques in static code analysis to detect weaknesses and vulnerabilities. In: Sokolova, M., van Beek, P. (eds.) Proceedings of Canadian Conference on AI'14. LNAI, vol. 8436, pp. 326–332. Springer (May 2014), short paper
13. Mokhov, S.A., Paquet, J., Debbabi, M.: MARFCAT: Fast code analysis for defects and vulnerabilities. In: Baysal, O., Guerrouj, L. (eds.) Proceedings of SWAN'15. pp. 35–38. IEEE (Mar 2015)
14. Mokhov, S.A., Song, M., Suen, C.Y.: Writer identification using inexpensive signal processing techniques. In: Sobh, T., Elleithy, K. (eds.) Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09. pp. 437–441. Springer (Dec 2009), ISBN: 978-90-481-9111-6, online at: `http://arxiv.org/abs/0912.5502`
15. Okun, V., Delaitre, A., Black, P.E., NIST SAMATE: Static Analysis Tool Exposition (SATE) IV. [online] (Mar 2012), see `http://samate.nist.gov/SATE.html`
16. The MARF Research and Development Group: The Modular Audio Recognition Framework and its Applications. [online] (2002–2014), `http://marf.sf.net` and `http://arxiv.org/abs/0905.1235`, last viewed May 2015