Table of Contents

Table of Contents **2**

Documentation Information **6**

Executive Summary **7**

A1: Zooniverse Showcase - Validation per Test Plan **10**
- Goals of Test Plan 10
- Test Implementation Plan 10

A2: Zooniverse Showcase - Validation Report **11**
- Technical performance 11
- Usability 11
- Overview of Test Results 11
- Detailed Test Reports 13
  - Background of TE-202: Animal and Emptiness Detection 13
    - Evaluation Approach 14
  - Test Framework Used 15
- Detailed Test Reports for the YOLO extractor 16
  - Test Sets Used 16
    - Entire Dataset 16
  - TP-202-01 - Test emptiness detection across a season of subjects 16
    - Metrics for blank test sets 17
    - Metrics for test sets with no blank images 17
    - Results 17
    - Overall 18
    - Simple vs Complex 18
    - Species Mix 18
    - Specific Species 18
- Detailed Test Reports for the DPM extractor 19
  - Test Sets Used 19
    - Entire Dataset 19
    - Complexity 20
    - Animal Size 20
    - Mixed Animal Types 20
    - Day and Night 20

TP-202-01 - Test emptiness detection across a season of subjects 21
Metrics for blank test sets 21
Metrics for test sets with no blank images 21
Results 22
Overall 22
Day vs Night 23
Simple vs Complex 23
Species Mix 23
Specific Species 23
TP-202-02 - Test animal type detection across a season of subjects 23
Metrics for test sets containing the desired species 23
Metrics for test sets which do not contain the desired species 24
Results: Buffalo 24
Results: Elephants 25
Results: Ostriches 26
Results: Warthogs 27
Results: Wildebeest 28
Results: Summary 29
Overall 29
Day vs Night 30
Simple vs Complex 30
Species Mix 30
Specific Species 30
TP-202-03 - Test animal counting across a season of subjects 30
Metrics for test sets with a mix of animal counts 30
Metrics for test sets where the desired count is impossible 31
Results: 1 animal 32
Results: 2 animals 32
Results: 3 animals 33
Results: 4 animals 34
Results: 5 animals 35
Results: 6 animals 36
Results: 7 animals 37
Results: 8 animals 38
Results: 9 animals 38
Results: 10 animals 40
Results: Summary 40
Overall 41
Day vs Night 41

Simple vs Complex 42
Species Mix 42
Specific Species 42
Overall Analysis of Platform for TP-202 (Animal Detection) 42
Latency 42
Throughput 43
Stability 43
Modularity 43
Integration 43
Usefulness 43
Evaluation of Testing Approach for TE-202 (Animal detection) 44
Background and Context of TE-506: Cross modal content recommender 45
TP-506-01 - Test that species preferences match user behaviour 46
TP-506-02 - Test that the subjects recommended match the preferred species 49
TP-506-03 - Test that different species preferences result in the correct changes to subject recommendations 51
TP-506-04 - Real-world tests using recommender results 51
TP-508-01 - Test subjects recommended 52
TP-508-02 Real-world tests using recommender results 54

B1: Use Cases: First Prototype - Video News Showcase - Test Plan 55
Goals 55
Test Implementation Plan 55

B2: Use Cases: First Prototype - Video News Showcase - Validation Report 57
Overview of Test Results 57
Details Test Reports 58
Background for TP-204-01 and TP-204-02: Face detection 60
Results Summary 62
BetaFaceAPI 62
MICO 62
Test Framework Used 64
TP-204-01 - Test in-front face detection of images 64
Metrics for the in-front face detection set 65
Results from MICO 65
Results from BetaFaceAPI 66
TP-204-02 - Test lateral face detection of images 66
Metrics for the side face detection set 66
Results from MICO 66
Results from BetaFaceAPI 66

Background for TP-214-01: Automatic Speech Recognition 66
Metrics for testing ASR 67
TP-214-01 - Test ASR on videos containing voiceover in English without noise and/or music 67
Test results 68
Latency 69
Throughput 69
Stability 69
Modularity 69
Integration 69
Usefulness 71

TP-WP05 - Item and User based use case tests 72
Greenpeace Magazine - Dataset Description 73
Greenpeace Magazine - Recommendation Evaluation 75
Zooniverse User Likes - Dataset Description and Recommendation Evaluation 77

WP2 orchestration - qualitative validation aspects 79

Documentation Information

Project (Title/Number)	MICO - ñMedia in Contextò(610480)
Work Package / Task	Work Package 7 - Use Case: Crowd Sourcing Platform Work Package 8 - Use Case: Video Sharing Platform
Responsible person and project partner	Chris Lintott (University of Oxford - Zooniverse) Andrea Volpini (InsideOut10)

Copyright
This document contains material, which is the copyright of certain MICO consortium parties, and may not be reproduced or copied without permission. The commercial use of any information contained in this document may require a license from the proprietor of that information. Neither the MICO consortium as a whole, nor a certain party of the MICO consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

Neither the European Commission, nor any person acting on behalf of the Commission, is responsible for any use which might be made of the information in this document.

The views expressed in this document are those of the authors and do not necessarily reflect the policies of the European Commission.

Executive Summary

This document outlines the Final Evaluation of the two use cases (WP7 and WP8) and the relevant functionalities that have been evaluated during the development of the MICO platform, including selected WP2 extractors, WP5 reco, and WP2 broker functionalities.

For the relevant MICO Technology Enablers (TE) tested, we assess accuracy, technical performance and/or usability in terms of integration, modularity and usefulness.

For **Zooniverse**, testing focused on evaluation of the animal/emptiness detectors via the MICO Platform API (TE-202), and on validation of the user profile generator and subject recommender first through functional testing and later through using them in an experimental scenario (TE-508).

TE-202 concerning animal/emptiness detection was tested using a test harness to process 50,270 images through the platform with the DPM extractor, 9869 images through the platform with the YOLO extractor, and collect and analyse the results, specifically in terms of Precision, Recall and F1 Score. We had originally planned to analyse 300,000 images but due to delays and technical issues with the deployed platform instance we had to use a reduced dataset. Findings were that the MICO platform is very precise at emptiness detection (92%), and has a great recall metric (99%), although admittedly this was calculated on a small sample size. This result is immediately useful for Zooniverse and can save several days of volunteer effort on a project. The platformôs performance for animal counting and species identification was less successful. We saw high recall for ostrich detection and high precision for wildebeest detection, however we suspect these results are caused by the prevalence of wildebeest and the rarity of ostriches in the dataset. Emptiness detection was more precise in the daytime, but animal counting was more precise at night. The complexity of an image, and the mix of species present, did not make much impact on the detectors, though multi-species images generally performed slightly better, perhaps benefitting from the input of multiple detectors. In this report we also analyse the testing approach used so that we can improve our approach next time.

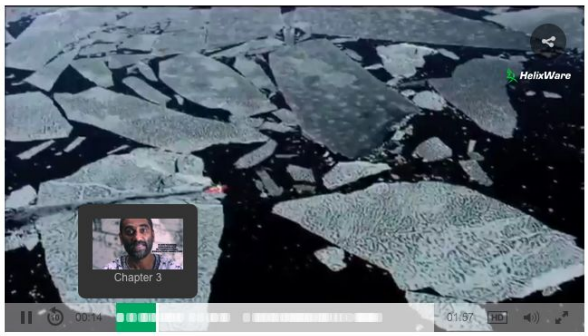
TE-508 concerning profile generation and subject recommendation did not proceed as planned, as the Zaizi solution did not materialise. However a script based approach was used by Zooniverse to generate subject content and species preference data for user profiles. This approach is validated and detailed below. The user profiles and subject recommendation routines were used in an experimental scenario, the ñHappy User Experimentò, which is detailed on the MICO blog. This work made a significant infrastructure contribution to Zooniverse, as Zooniverse now has a user behaviour collector and experiment management system that can be used in future experiments. The experiment found that providing users with ñpreferredò species did not encourage greater participation, in fact it deterred users and they left sooner. This was investigated further and lead to new findings that the impact of blank images in a

users's image set can have a strong impact on engagement. These findings were published at HCOMP 2015 and further details and links are provided below.

For **InsideOut10** we have kept the scope of the validation as previously defined, *not* including

- media quality TE-205, for which integration has been discontinued during y2
- temporal video segmentation TE-206, which is available within the system, and we've updated for this year the components to reflect the changes in the data model and which has been successfully integrated in a live scenario, but is not part of this evaluation

Hence, the focus has been on two extractors, TE-204 face detection and TE-214 automatic speech recognition:



The new HelixWare player with chapters

- A controlled evaluation of the results is made in terms of precision, recall and F1 measure conducted in our lab without engaging (other than for gathering the datasets and sharing the results) our stakeholders Greenpeace and other users of the WordLift platform.
- A limited amount of data is analyzed and processed (60 images and 10 videos)
- There is a specific Greenpeace use-case for WP5 and an *internal evaluation*. The required functionality is built in the recommendation modules of the Mico platform - unfortunately we could not complete the A/B testing with our test group as expected. Therefore a preliminary offline study was conducted with the support of the team from Fraunhofer. The direction here is taken and the work has been progressing regardless of the changes in the leadership team of this work package. As now - at least for our showcase - with the work done on prediction.io there seems to be a valuable approach to cross-media recommendations.

On the positive side, we can confirm that face detection continues in providing results of immediate use and overall comparable with the results of the analysis offered by commercial service providers (this is particularly true for in-front face detections). This is indeed a great outcome for MICO and a clear evidence that the work done on the extractors is consistent and *state-of-the-art*. While in this past year there have not been major changes on the extractor regarding the IO10 showcase, we have seen the expected flexibility in terms of pipeline configuration that the new broker assures.

We have also seen - as in the previous evaluation - the value of orchestrating different extractors when for instance analyzing audio-visual contents: a face detection running after the temporal video segmentation is definitely more useful - while harder to evaluate in terms of precision/recall - than running the extractor by itself on all image stills (when bringing the results of the analysis back to the end-user the application developer needs to control and limit the amount of information provided via the user interface).

Regarding TE-214 (automatic speech recognition), automatically splitting a video in audio and video parts is quite effective in terms of workflow, but unfortunately the freely accessible language models cannot compete with commercially available ones.

This work also serve as a good base for integrators working in *JAVA* (the code has been contributed as open source and it is available on GitHub). The infrastructure developed is also intended for the vast WordPress community (27% of all existing websites according to the latest W3Techs report) interested in media analysis and recommendations.

Last but not least we have build a tool for evaluating the results of the analysis of MICO on images and for creating the ground data (manual annotations) - this methodology and approach can be re-used for future testing.

The report also summarizes some results for evaluating **WP5 recommendation**. Due to the fact that respective usage data was limited, and that WP5 depended on the *complete* platform and all relevant extractos to be completed first, evaluation had to be done in significantly less time and a more limited scope than originally planned. However, several interesting insights could be derived, and with the framework available now, future testing can now be done with significantly lowered effort.

For **WP2 orchestration**, the report qualitatively compares the results achieved with broker v3 with the broker wishlist that was compiled in y2. From that, it is clear that most goals could be achieved, and several more goals (some of them not yet foreseen in y2) could now be achieved with a limited investment.

A1: Zooniverse Showcase - Validation per Test Plan

Goals of Test Plan

The goals of the test plan were stated as follows:

We aim to compare each MICO TE prior to beginning end-to-end testing of the system. For each MICO TE included in these tests we will assess the following:

1. **output accuracy** - how accurate, detailed and meaningful each single response is when compared to our best estimates using our existing databases and analysis;
2. **technical performance** - how much time each task requires and how scalable the solution is when we increase in volume the amount of contents being analysed;
3. **usability** evaluated both in terms of **integration, modularity** and **usefulness**.

It was noted in the test plan, and we note again here, that a low score on these metrics does not indicate a failure. It is important to collect these metrics to correctly understand what the system does and does not do - but this does not constitute a success/fail judgement of the platform as a whole.

Test Implementation Plan

To recap, the test plan for Zooniverse was as follows:

ID	Test	Description
TP-202-01	**Test emptiness detection across a season of subjects**	ŏ use dataset **TD-12** ŏ process **TD-12** with TE-202 (emptiness detector) ŏ calculate precision, recall and F1-measure on **TE-202** outputs. ŏ calculate manual blanks, for **TD-12** per normalization procedure. ŏ calculate precision, recall and F1-measure on outputs. ŏ compare **KPIs**
TP-202-02	**Test animal type detection across a season of subjects**	ŏ use dataset **TD-12** ŏ process **TD-12** with TE-202 (group detector) ŏ calculate precision, recall and F1-measure on **TE-202** outputs. ŏ calculate manual animal types present, for **TD-12** per normalization procedure. ŏ calculate precision, recall and F1-measure on outputs. ŏ compare **KPIs**
TP-202-03	**Test animal counting across a season of subjects**	ŏ use dataset **TD-12** ŏ process **TD-12** with TE-202 (animal detector) ŏ calculate precision, recall and F1-measure on **TE-202** outputs. ŏ calculate manual animal counts for **TD-12** per normalization procedure. ŏ calculate precision, recall and F1-measure on outputs. ŏ compare **KPIs**

TP-508-01	**Test subjects recommended**	ŏ use dataset **TD-12 and TD-20** ŏ Generate TD-22 by processing TD-12 and TD-20. ŏ calculate precision, recall and F1-measure on **TE-508** outputs. ŏ Process results from TE-508 per discussion against TD-21, and identify discussions that were recommended for each given discussion. ŏ Compare discussion recommendation to TD-21. ŏ Assess KPIs
TP-508-02	**Real-world tests using recommender results**	ŏ use dataset **TD-11** ŏ Run a full experiment over a period of days/weeks, showing recommended discussion threads to users. ŏ Evaluate click-through rate and explicitly reported ñhelpfulnessò rating. ŏ Determine whether the recommended discussion threads increased or decreased user participation. ŏ assess **KPIs**

A2: Zooniverse Showcase - Validation Report

The following metrics will be used:

Technical performance

Technical performance will be measured in terms of:

- ŏ **latency** - time required to perform a single task on a given dataset. Measures will be repeated 10 times;
- ŏ **scalability** - an assessment of whether the given TE is suitably efficient and practical when applied to large situations (e.g. a large input dataset and / or, a large number concurrent requests)

Usability

The TE usability requires a qualitative evaluation which will consider:

- ŏ **integration** - how simple it is to integrate each single TE into Zooniverse technologies;
- ŏ **modularity** - how simple it is to configure each TE and/or a combination of multiple TEs in a chain from within pre-existent application workflows.
- ŏ **usefulness** - looking at the degree to which the TE delivers valuable information and tools that Zooniverse applications will be able to harness in future, and some consideration towards a cost-benefit analysis of doing so.

Overview of Test Results

ID	Test	Summary of Results
TP-202-01	**Test emptiness detection across a season of subjects**	ŏ Overall, the YOLO extractor improved greatly on the DPM extractor. ŏ YOLO precision was higher (92%, compared to 84% for DPM) and recall was much better (99%, compared to 45% for DPM)

		ð We see a tendency to over classify in both extractors, which for the Zooniverse use case is actually preferable over false negatives. ð MICO seems to be much more precise at determining blanks during the day than at night. Recall doesn't vary much between day and night. ð Most importantly, **this new capability to detect blanks is immediately useful to Zooniverse, and can save Zooniverse over 115,000 classifications on a 300,000 image dataset - several days to a week of volunteer effort.**
TP-202-02	**Test animal type detection across a season of subjects**	**DPM extractor:** ð Precision was very low (around 6%), though about 10% better for wildebeest. ð Ostriches achieved a high recall (61%) though we think this is solely due to overdetection of ostriches across all images combined with ostriches being very rare in the dataset. ð Wildebeest detection seemed to perform much better (69% precision) in complex images than in simple images (30% precision). We think this is explained by overdetection of wildebeest across all images combined with wildebeest being especially prevalent in the dataset.
TP-202-03	**Test animal counting across a season of subjects**	**DPM extractor:** ð Both precision and recall were uniformly very low. The MICO platform is very poor at being correct about the number of animals when it does detect animals, and very likely to detect an incorrect number of animals. ð There is some evidence to suggest that MICO is more precise at counting animals in nighttime shots, at least for images containing a single animal - perhaps because distracting features such as clouds, bushes and shadows are not visible, reducing false positives. ð Images with more than one species performed slightly better, perhaps because multiple MICO detectors contributed to the decision. ð When testing images containing a single animal, the detectors for warthogs and ostriches performed especially well (precision of 75-84%) and the detector for wildebeest performed especially poorly (precision of 21%). These results may be partly explained by the prevalence of wildebeest and the rarity of ostriches in the dataset.
TP-506-01	**Test that species preferences match user behaviour***	ð Simple queries sufficed to run the real-world tests, a machine learning approach was not needed and was not developed. ð Species preferences as it came out of those queries matched user behaviour in all cases. ð However, that means KPI values could not be calculated since there is no further benchmark to compare against.
TP-506-02	**Test that the subjects recommended match the preferred species***	ð Simple queries sufficed to run the real-world tests, a machine learning approach was not needed and was not developed. ð Subjects recommended as they came out of those queries contained the preferred species in all cases. ð However, that means KPI values could not be calculated since there is no further benchmark to compare against

TP-506-03	**Test that different species preferences result in the correct changes to subject recommendations***	ǒ Simple queries sufficed to run the real-world tests, a machine learning approach was not needed and was not developed. ǒ Subjects recommended were all updated correctly when species preferences are updated and the recommendations recalculated. ǒ However, that means KPI values could not be calculated since there is no further benchmark to compare against.
TP-506-04	**Real-world tests using recommender results**	ǒ The user profile data enabled us to run the ñHappy User Experimentò This experiment told us that adding more interesting animal images actually deters users, contrary to our expectations. The finding is written up in more detail on the MICO blog: http://www.mico-project.eu/snapshot-serengeti-an-unexpected-discovery/ ǒ This experiment also opened up new avenues of research: ǯ http://bit.ly/blanks-poster ǯ http://bit.ly/blanks-papers ǒ .. and made significant contributions to build infrastructure that Zooniverse can re-use in future: ǯ https://github.com/zooniverse/geordi ǯ https://github.com/zooniverse/experiment-server ǯ https://github.com/zooniverse/geordi-client
TP-508-01	**Test subjects recommended**	ǒ Software delivered too late for full evaluation ǒ Initial findings are very positive ǒ Found images where our aggregated answers from the crowd had missed some of the animals
TP-508-02	**Real-world tests using recommender results**	Due to the time the WP5 recommendation engine was released, there was not enough time left in the project to run this planned evaluation.

Detailed Test Reports

Background of TE-202: Animal and Emptiness Detection

The Fraunhofer team developing the extractors decided that initially, the focus of development effort, and thus testing, would be on the following species:

- ǒ elephant
- ǒ ostrich
- ǒ warthog
- ǒ wildebeest
- ǒ buffalo

These species were selected as those where the team were most able to deliver a guess with acceptable accuracy using the planned detection approach.

After year two, we found that the best results were with the distinction between images with animals in them, and images without any animals at all. This is also easily usable for the Zooniverse, because we can use that to in the decision when we can stop showing images to volunteers.

However, while the DPM-based extractor yielded usable results, the Fraunhofer team felt it could improve on it with a new YOLO-based extractor. The YOLO extractor is much slower at processing images, which is why we decided to focus YOLO reporting on how much of an improvement it was for blank image detection. A more extensive report comparing all the aspects using specific subsets of processed images was unfeasible in the timeframe we had.

Evaluation Approach

In order to assess the accuracy of each MICO platform determination, we need to know what is in each image. Fortunately, through human computation on Snapshot Serengeti, we have a ñcrowd answerò for every one of the 300,000 images in TD-12. As planned in ñDataset Normalizationòin the Test Plan, we used aggregation techniques to generate an easy-to-use summary for each image, including what species it contained, how many animals were present, what time of day the image was taken, whether it was blank, and other data.

This was done using the script:
https://github.com/zooniverse/mongo-subject-extractor/blob/master/generate_detailed_consensus.rb

which generated a CSV which is available here:
https://github.com/zooniverse/mongo-subject-extractor/raw/master/consensus-detailed.csv.zip

and which looks like this (simplified for readability):

```
zooniverse_id,season,site_id,frames,time_of_day,classifications,crowd
_says,total_species,total_animals,crowd_says_if_multi,retire_reason
ASG000tu6x,5,U12,3,11:41,12,zebra,1,5,zebra,consensus
ASG000tu6y,5,U12,3,11:42,10,zebra,1,5,zebra,consensus
ASG000tu6z,5,U12,3,11:43,17,multi,2,6,wildebeest;zebra,consensus
```

This CSV was imported into the test harness described below, and used for evaluation of MICO results.

Test Framework Used

We used the MICO Platform API developed by the team at Salzburg Research. See http://mico-project.bitbucket.org/api/ or document D6.2.2 for further details of custom animal detection and text analysis endpoints.

Using this API, the Zooniverse team was able to develop a test harness. The test harness is built into our MICO demo, the code for which is at:
https://github.com/zooniverse/mico-serengeti-demo

The demo is currently deployed at:
http://mico-demo.snapshotserengeti.org/subjects)
The testing logic (which can be run as described below) had the following approach:

1. Run all images in entire_dataset through the API, with the platform set to animal detection mode
2. Keep polling the API until all images are processed (status=ðinishedð)
3. Store the MICO results in a database.
4. Calculate KPI test results and export to CSV:

RESULTS GENERATION SCRIPT - KPI Calculation - Pseudocode

```
for each defined test
  for each defined test set
    calculate the sample size
    for each subject
      compare each subject in this set against the subject content
CSV
      calculate whether true positive / false negative / etc
    next subject
    analyse total, calculate the KPI: precision, recall and F1 score
    generate a CSV row summarising results for this test set & test
  next test set
  generate summary CSV row for this test
next test
output all results to CSV
```

5. The CSV data (a sample of which is shown here, numbers rounded for readability) was then analysed to produce this report.

```
detector, filter,    sample_size,true_pos,true_neg,false_pos,false_neg,precision,recall,f1
emptiness,everything,50270,      16425,   10451,   3085,     20309,    0.84,     0.45,  0.58
emptiness,daytime,   368,        106,     76,      14,       172,      0.88,     0.38,  0.53
emptiness,nighttime, 183,        33,      76,      45,       29,       0.42,     0.53,  0.47
```

The results generation script is available on Github. See the demo projectôs README on how to use it.

Detailed Test Reports for the YOLO extractor

We ran our validation on a subset of TD-12. Because YOLO is much slower to run than the DPM extractor, we had about 1% of the entire dataset run through the extractor pipeline, or 10,000 images.

Test Sets Used

Under the test plan we had expected to use the TD-12, also known as Season 8, which contains 300,000 images. Unfortunately, as described above, we were only able to have a total of **9869 images** analysed by the platform. For the purposes of this document, we refer to this set as the ñentire datasetò- though in future we will expand testing with the full 300,000. Fortunately we were able to ensure good coverage for the five key species within this 50,270 images.

Entire Dataset

Each part of the dataset is given a test set name or filter name, and these are marked in brackets. The name for the whole set is **[ñentire_datasetò]**, which breaks down as follows:

- ŏ 7828 blank images **[ñblankò]**
- ŏ 1940 images containing a single species **[ñsingle_speciesò]**, consisting of:
  - ʒ 51 images containing only buffalo **[ñonly_buffaloò]**
  - ʒ 54 images containing only elephant(s) **[ñonly_elephantò]**
  - ʒ 46 images containing only warthog(s) **[ñonly_warthogò]**
  - ʒ 670 images containing only wildebeest* **[ñonly_wildebeestò]**
  - ʒ 5 images containing only ostrich(es)** **[ñonly_ostrichò]**
  - ʒ 1114 images containing only a single species, not one of the above **[ñonly_otherò]**
- ŏ 101 images containing a mixture of species **=> ñmulti_speciesò.** 73 of these images contained at least a buffalo. The other species were negligibly represented.

* : This data set was a much larger than the others because wildebeest are especially common in all Serengeti images

** : This data set was a much smaller than the others because ostrich are especially rare in all Serengeti images. 357 represents the total number of ostrich images available in the 300,000 original images.

TP-202-01 - Test emptiness detection across a season of subjects

For this test, we wanted to evaluate how the MICO platform performs at detecting whether or not an image contains animals. This corresponds to the Accuracy Definition for ñEmptinessò under Methodology and Planning in the Test Plan.

Metrics for blank test sets

For testing the correct detection of an empty image, we defined a single test, **emptiness**, the metrics for which are as follows:

- Relevance is defined as ñThe image is blankò
- Selected is defined as ñThe image is identified as blankò
- This leads to the following definitions:
  - True positive (TP): Blank image correctly identified as blank
  - True negative (TN): Non-blank image correctly identified as containing animals
  - False negative (FN): Blank image incorrectly identified as containing animals
  - False positive (FP): Non-blank image incorrectly identified as blank
  - Precision: `TP / (TP+FP)` : The percentage of images identified as blank which do indeed contain no animals.
  - Recall: `TP / (TP+FN)` : The percentage of images which are actually blank that are correctly identified as containing no animals.
  - F1 score: `2*(Precision*Recall)/(Precision+Recall)` : overall measure of this testôs accuracy, combining precision and recall

Metrics for test sets with no blank images

A number of the test sets, by definition, contained no blank images. Therefore for these tests, rather than omitting the test, we looked at how good the detector was at detecting the *absence* of empty images. In other words, we used an alternate definition of relevance in these tests:

- Relevance is defined as ñThe image contains one or more animalsò
- Selected is defined as ñThe image is identified as containing no animalsò
- This leads to the following definitions:
  - True positive: Non-blank image correctly identified as containing animals
  - True negative: Not applicable in this case, as there are no blank images.
  - False negative: Non-blank image incorrectly identified as containing no animals
  - False positive: Not applicable in this case, as there are no blank images.
  - Precision: Not applicable in this case, as there are no non-relevant images.
  - Recall: The percentage of images which are actually contain animals that are correctly identified as containing animals.
  - F1 score: Not applicable in this case, as there is no figure for precision.

Test sets where this alternate set of metrics was used are marked with an asterisk in the results table below.

Results

The results were as follows. In this and all subsequent results tables, precision, recall and F1 scores of 75% or better are highlighted in green, and those same metrics are highlighted in red when they are equal to 25% or less.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	9869	7828	7746	1410	631	82	92%	99%	96%
blank	7828	7828	7746	0	0	82	N/A	99%	N/A
non_blank	2041	2041	0	1410	631	0	N/A	69%	N/A
one_animal	813	813	0	499	314	0	N/A	61%	N/A
simple	705	705	0	532	173	0	N/A	75%	N/A
complex	523	523	0	379	144	0	N/A	72%	N/A
single_species	1940	1940	0	1336	604	0	N/A	69%	N/A
only_buffalo	51	51	0	36	15	0	N/A	71%	N/A
only_elephant	54	54	0	20	34	0	N/A	37%	N/A
only_ostrich	5	5	0	1	4	0	N/A	20%	N/A
only_warthog	46	46	0	41	5	0	N/A	89%	N/A
only_wildebeest	670	670	0	436	234	0	N/A	65%	N/A
only_other	1114	1114	0	802	312	0	N/A	72%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets with no blank imagesò above.

Overall

Precision was high: When YOLO determines an image to be blank, it has a high likelihood (92%) of being right.
Recall was extremely high: MICO falsely classified just 1% of blank images as containing animals.

Simple vs Complex

The number of animals in an image makes some difference in getting it right. Images with more than one animal were about 11-14% more likely to be correctly identified as non-blank than those with one animal. It seems like there is also a sweet spot, YOLO was better at classifying images with a few animals than it was at lots of animals. Intuitively this seems to make sense: a few animals gives more chances of finding at least one, many animals means more overlapping animals which is harder on the detector.

Species Mix

The number of different species in an image did not make much difference in the ability of YOLO to be sure that it wasnôt blank. Images with more than one species were about 4% more likely to be correctly identified as non-blank than those with one single species. This is probably because the results of multiple different species detectors contributed to the answer.

Specific Species

YOLOôs ability to confirm that images containing each of the different species was non-blank varied between species. In particular, elephants (recall of 37%) and ostriches (recall of 20%) were much lower than the other species. Warthogs performed notably the best at 89% recall. The other species were all in the 70% range.

Detailed Test Reports for the DPM extractor

We ran our validation on a subset of TD-12. In total, we had about 25% of the entire dataset run through the DPM extractor pipeline, or 250,000 images. Of this 25%, only 46,321 images had any usable result. No response was received for the other images.

Due to platform availability, we were only able to test an overall dataset of 50,270 images rather than the planned ~300,000. Nonetheless, we have been able to evaluate the performance, accuracy and usability of the platform.

Test Sets Used

Under the test plan we had expected to use the TD-12, also known as Season 8, which contains 300,000 images. Unfortunately, as described above, we were only able to have a total of **50,270 images** analysed by the platform. For the purposes of this document, we refer to this set as the ñentire datasetò- though in future we will expand testing with the full 300,000. Fortunately we were able to ensure good coverage for the five key species within this 50,270 images.

Entire Dataset

Each part of the dataset is given a test set name or filter name, and these are marked in brackets. The name for the whole set is **[ñentire_datasetò]**, which breaks down as follows:

- 36,734 blank images **[ñblankò]**
- 13,045 images containing a single species **[ñsingle_speciesò]**, consisting of:
  - 1,169 images containing only buffalo **[ñonly_buffaloò]**
  - 1,190 images containing only elephant(s) **[ñonly_elephantò]**
  - 1,1090 images containing only warthog(s) **[ñonly_warthogò]**
  - 3,057 images containing only wildebeest* **[ñonly_wildebeestò]**
  - 356 images containing only ostrich(es)** **[ñonly_ostrichò]**
  - 6,183 images containing only a single species, not one of the above **[ñonly_otherò]**
- 491 images containing a mixture of species **=> ñmulti_speciesò.** These can be filtered as follows (noting that some images occur in more than one group):
  - 18 images containing buffalo along with other species **[ñmulti_including_buffaloò] *****
  - 7 images containing elephant(s) along with other species **[ñmulti_including_elephantò] *****

- 25 images containing warthog(s) along with other species **[ñmulti_including_warthogò] *****
- 323 images containing wildebeest along with other species **[ñmulti_including_wildebeestò]**
- 1 image containing ostrich(es) along with other species **[ñmulti_including_buffaloò] *****
- 124 images containing none of the above species **[ñmulti_including_none_of_the_fiveò]**

* : This data set was much greater than the others because wildebeest are especially common in all Serengeti images
** : This data set was much smaller than the others because ostrich are especially rare in all Serengeti images. 357 represents the total number of ostrich images available in the 300,000 original images.
*** : We have decided not to use these subsets in tests, as they have a sample size of less than 100 and therefore the results would not be useful or meaningful.

Complexity

To address the borderline case of simple versus complex, which is measured by the number of actual animals (regardless of species) present, we also broke down the sum of **single_species** and **multi_species** (in other words all non-blank images) as follows:

- 13,536 non-blank images **[ñnon_blankò]**, consisting of:
  - 6,172 images with only 1 animal present **[ñone_animalò]**
  - 4,583 images with between 2 and 5 animals present **[ñsimpleò]**
  - 2,781 images with 6 or more animals present **[ñcomplexò]**

Animal Size

Given the fact that only 5 species were chosen with a variety of sizes between them (only buffalo and wildebeest share a similar size), and given that the detectors no longer deal in animal groups but rather in single species, we decided it was not necessary to test borderlines cases based on animal size. This will be addressed in future at the point where we support multiple species of each size.

Mixed Animal Types

To address the borderline case of single species images versus images containing a mixture of species, we used the **single_species** and **multi_species** groups as defined above (and their subgroups).

A note on animal types: The original test plan envisaged us that the MICO platform would recognize groups of species e.g. cat-like animals. Since this approach has been abandoned, we no longer worry about animal types and instead focus on the specific five species.

Day and Night

To test whether there is a difference in performance between daytime images and nighttime images, we created two small test sets of solely daytime and solely night time images.

Unfortunately, these are quite small, because timestamps in the data prove to be unreliable, therefore we had to create these sets by visual inspection. The technical problem of how to accurately discern day and nighttime images without having to visually inspect every image is left as a possible future exercise.

The test sets here were:

- 368 images taken during daylight hours **[ñdaytimeò]**
- 183 images taken during hours of darkness **[ñnighttimeò]**

TP-202-01 - Test emptiness detection across a season of subjects

For this test, we wanted to evaluate how the MICO platform performs at detecting whether or not an image contains animals. This corresponds to the Accuracy Definition for ñEmptinessò under Methodology and Planning in the Test Plan.

Metrics for blank test sets

For testing the correct detection of an empty image, we defined a single test, **emptiness**, the metrics for which are as follows:

- Relevance is defined as ñThe image is blankò
- Selected is defined as ñThe image is identified as blankò
- This leads to the following definitions:
  - True positive (`TP`): Blank image correctly identified as blank
  - True negative (`TN`): Non-blank image correctly identified as containing animals
  - False negative (`FN`): Blank image incorrectly identified as containing animals
  - False positive (`FP`): Non-blank image incorrectly identified as blank
  - Precision: `TP/(TP+FP)` : The percentage of images identified as blank which do indeed contain no animals.
  - Recall: `TP/(TP+FN)` : The percentage of images which are actually blank that are correctly identified as containing no animals.
  - F1 score: `2*(Precision*Recall)/(Precision+Recall)` : overall measure of this testôs accuracy, combining precision and recall

Metrics for test sets with no blank images

A number of the test sets, by definition, contained no blank images. Therefore for these tests, rather than omitting the test, we looked at how good the detector was at detecting the *absence* of empty images. In other words, we used an alternate definition of relevance in these tests:

- Relevance is defined as ñThe image contains one or more animalsò
- Selected is defined as ñThe image is identified as containing no animalsò
- This leads to the following definitions:
  - True positive: Non-blank image correctly identified as containing animals
  - True negative: Not applicable in this case, as there are no blank images.
  - False negative: Non-blank image incorrectly identified as containing no animals
  - False positive: Not applicable in this case, as there are no blank images.

- ð Precision: Not applicable in this case, as there are no non-relevant images.
- ð Recall: The percentage of images which are actually contain animals that are correctly identified as containing animals.
- ð F1 score: Not applicable in this case, as there is no figure for precision.

Test sets where this alternate set of metrics was used are marked with an asterisk in the results table below.

Results

The results were as follows. In this and all subsequent results tables, precision, recall and F1 scores of 75% or better are highlighted in green, and those same metrics are highlighted in red when they are equal to 25% or less.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	36,734	16,425	10,451	3,085	20,309	84%	45%	58%
daytime	368	278	106	76	14	172	88%	38%	53%
nighttime	183	62	33	76	45	29	42%	53%	47%
blank	36,734	36,734	16,425	0	0	20,309	N/A	45%	N/A
non_blank	13,536	13,536*	10,451	0	0	3,085	N/A	77%	N/A
one_animal	6,172	6,172*	4,586	0	0	1,586	N/A	74%	N/A
simple	4,583	4,583*	3,674	0	0	909	N/A	80%	N/A
complex	2,781	2,781*	2,191	0	0	590	N/A	79%	N/A
single_species	13,045	13,045*	10,042	0	0	3,003	N/A	77%	N/A
only_buffalo	1,169	1,169*	913	0	0	256	N/A	78%	N/A
only_elephant	1,190	1,190*	893	0	0	297	N/A	75%	N/A
only_ostrich	356	356*	261	0	0	95	N/A	73%	N/A
only_warthog	1,090	1,090*	709	0	0	381	N/A	65%	N/A
only_wildebeest	3,057	3,057*	2,512	0	0	545	N/A	82%	N/A
only_other	6,183	6,183*	4,754	0	0	1,429	N/A	77%	N/A
multi_species	491	491*	409	0	0	82	N/A	83%	N/A
multi_including_ wildebeest	323	323*	267	0	0	56	N/A	N/A	N/A
multi_including_ none_of_the_five	124	124*	107	0	0	17	N/A	N/A	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets with no blank imagesòabove.

Overall

Precision was high: When MICO determines an image to be blank, it has a high likelihood (84%) of being right.
Recall was low: MICO falsely classified 55% of blank images as containing animals. This is not a bad thing from Zooniverseôs perspective, the tendency to over classify is useful in ensuring we do not miss anything.

Day vs Night

MICO seems to be much more precise at determining blanks during the day than at night. At night, it is twice as likely to falsely consider a non-blank image to be blank.
The recall was similar between day and night - night time blank images are not noticeably harder to determine as blank as far as MICO is concerned

Simple vs Complex

The number of animals in an image did not make much difference in the ability of MICO to be sure that it wasnôt blank. Images with more than one animal were about 5-6% more likely to be correctly identified as non-blank than those with one animal.

Species Mix

The number of different species in an image did not make much difference in the ability of MICO to be sure that it wasnôt blank. Images with more than one species were about 6% more likely to be correctly identified as non-blank than those with one single species. This is probably because the results of multiple different species detectors contributed to the answer.

Specific Species

MICOôs ability to confirm that images containing each of the different species was non-blank did not vary widely between species. Warthogs performed slightly worse, at 65% recall, and wildebeest slightly better, at 83%. The other species were all in the 70% range.

TP-202-02 - Test animal type detection across a season of subjects

For this test, we wanted to evaluate how the MICO platform performs at detecting the specific species of elephant, buffalo, wildebeest, warthog or ostrich. Or not an image contains animals. This corresponds to the Accuracy Definition for ñAnimal Typeò under Methodology and Planning in the Test Plan. (Recall that the platform did not train animals in ñtypeògroups as originally planned, so animal type simply maps to ñspeciesò

Metrics for test sets containing the desired species

For testing the correct detection of an empty image, we defined five tests, **animal_type_buffalo**, **animal_type_elephant**, **animal_type_warthog**, **animal_type_wildebeest**, and **animal_type_ostrich,** the metrics for which are as follows:

- Relevance is defined as ñThe image contains the specified speciesò
- Selected is defined as ñThe image is identified as containing the specified speciesò
- This leads to the following definitions, for a given species x:
    - True positive: Image containing species x correctly identified as containing species x

- ǒ True negative: Image that does not contain species x correctly identified as not containing species x
- ǒ False negative: Image containing species x incorrectly identified as not containing species x
- ǒ False positive: Image that does not contain species x incorrectly identified as containing species x
- ǒ Precision: The percentage of images identified as containing species x which do indeed contain species x.
- ǒ Recall: The percentage of images which actually contain species x that are correctly identified as containing species x.
- ǒ F1 score: overall measure of this testôs accuracy, combining precision and recall

Metrics for test sets which do not contain the desired species

A number of the test sets, by definition, did not contain the desired species for that test. Therefore for these tests, rather than omitting the test, we looked at how good the detector was at detecting the *absence* of species x. In other words, we used an alternate definition of relevance in these tests:

- ǒ Relevance is defined as ñThe image does not contain species xò
- ǒ Selected is defined as ñThe image is identified as not containing species xò
- ǒ This leads to the following definitions:
  - ǒ True positive: Image that does not contain species x correctly identified as not containing species x
  - ǒ True negative: Not applicable in this case, as there are no images containing species x.
  - ǒ False negative: Image that does not contain species x incorrectly identified as containing species x.
  - ǒ False positive: Not applicable in this case, as there are no images containing species x.
  - ǒ Precision: Not applicable in this case, as there are no non-relevant images.
  - ǒ Recall: The percentage of images which do not contain species x that are correctly identified as not containing species x.
  - ǒ F1 score: Not applicable in this case, as there is no figure for precision.

Test sets where this alternate set of metrics was used are marked with an asterisk in the results table below.

Results: Buffalo

The results for **animal_type_buffalo** were as follows.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	1,187	387	41,179	7,904	800	5%	33%	8%

daytime	368	3	0	308	57	3	0%	0%	N/A
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	32,091	0	0	4,643	N/A	87%	N/A
non_blank	13,536	1,187	387	9,088	3,261	800	11%	33%	16%
one_animal	6,172	464	146	4,280	1,428	318	9%	31%	14%
simple	4,583	415	138	2,965	1,203	277	10%	33%	16%
complex	2,781	308	103	1,843	630	205	14%	33%	20%
single_species	13,045	1,169	381	8,739	3,137	788	11%	33%	16%
only_buffalo	1,169	1,169	381	0	0	788	N/A	33%	N/A
only_elephant	1,190	1,190*	768	0	0	422	N/A	65%	N/A
only_ostrich	356	356*	288	0	0	68	N/A	81%	N/A
only_warthog	1,090	1,090*	819	0	0	271	N/A	75%	N/A
only_wildebeest	3,057	3,057*	2,111	0	0	946	N/A	69%	N/A
only_other	6,183	6,183*	4,753	0	0	1,430	N/A	77%	N/A
multi_species	491	18	6	349	124	12	5%	33%	8%
multi_including_ wildebeest	323	323*	237	0	0	86	N/A	73%	N/A
multi_including_ none_of_the_five	124	124*	90	0	0	34	N/A	73%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets which do not contain the desired speciesò above.

The tests using the nighttime test sets were not available for this test, as there were no images containing buffalo in this test set.

Results: Elephants

The results for **animal_type_elephant** were as follows.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	1,197	277	44,691	4,382	920	6%	23%	9%
daytime	368	8	1	326	34	7	3%	13%	5%
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	34,048	0	0	2,686	N/A	93%	N/A
non_blank	13,536	1,197	277	10,643	1,696	920	14%	23%	17%

one_animal	6,172	610	130	4,810	752	480	15%	21%	17%
simple	4,583	473	133	3,497	613	340	18%	28%	22%
complex	2,781	114	14	2,336	331	100	4%	12%	6%
single_species	13,045	1,190	275	10,213	1,642	915	14%	23%	18%
only_buffalo	1,169	1,169*	963	0	0	206	N/A	82%	N/A
only_elephant	1,190	1,190	275	0	0	915	N/A	23%	N/A
only_ostrich	356	356*	335	0	0	21	N/A	94%	N/A
only_warthog	1,090	1,090*	957	0	0	133	N/A	88%	N/A
only_wildebeest	3,057	3,057*	2,603	0	0	454	N/A	85%	N/A
only_other	6,183	6,183*	5,355	0	0	828	N/A	87%	N/A
multi_species	491	7	2	430	54	5	4%	29%	6%
multi_including_ wildebeest	323	323*	281	0	0	42	N/A	87%	N/A
multi_including_ none_of_the_five	124	124*	117	0	0	7	N/A	94%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets which do not contain the desired speciesò above.

The tests using the nighttime test sets were not available for this test, as there were no images containing elephants in this test set.

Results: Ostriches

The results for **animal_type_ostrich** were as follows.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	357	219	29,181	20,732	138	1%	61%	2%
daytime	368	2	2	192	174	0	1%	100%	2%
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	22,985	0	0	13,749	N/A	63%	N/A
non_blank	13,536	357	219	6,196	6,983	138	3%	61%	6%
one_animal	6,172	287	176	2,991	2,894	111	6%	61%	10%
simple	4,583	67	41	2,033	2,483	26	2%	61%	3%
complex	2,781	3	2	1,172	1,606	1	0%	67%	0%
single_species	13,045	356	219	6,018	6,671	137	3%	62%	6%

only_buffalo	1,169	1,169*	631	0	0	538	N/A	54%	N/A
only_elephant	1,190	1,190*	720	0	0	470	N/A	61%	N/A
only_ostrich	356	356	219	0	0	137	N/A	62%	N/A
only_warthog	1,090	1,090*	714	0	0	376	N/A	66%	N/A
only_wildebeest	3,057	3,057*	1,353	0	0	1,704	N/A	44%	N/A
only_other	6,183	6,183*	2,600	0	0	3,583	N/A	42%	N/A
multi_species	491	1	0	178	312	1	0%	0%	N/A
multi_including_ wildebeest	323	323*	116	0	0	207	N/A	36%	N/A
multi_including_ none_of_the_five	124	124*	43	0	0	81	N/A	35%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets which do not contain the desired speciesòabove.

The tests using the nighttime test sets were not available for this test, as there were no images containing ostriches in this test set.

Results: Warthogs

The results for **animal_type_warthog** were as follows.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	1,115	432	35,643	13,512	683	3%	39%	6%
daytime	368	9	3	250	109	6	3%	33%	5%
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	27,977	0	0	8,757	N/A	76%	N/A
non_blank	13,536	1,115	432	7,666	4,755	683	8%	39%	14%
one_animal	6,172	836	344	3,451	1,885	492	15%	41%	22%
simple	4,583	257	79	2,496	1,830	178	4%	31%	7%
complex	2,781	22	9	1,719	1,040	13	1%	41%	2%
single_species	13,045	1,090	422	7,384	4,571	668	8%	39%	14%
only_buffalo	1,169	1,169*	668	0	0	501	N/A	57%	N/A
only_elephant	1,190	1,190*	707	0	0	483	N/A	59%	N/A
only_ostrich	356	356*	261	0	0	95	N/A	73%	N/A
only_warthog	1,090	1,090	422	0	0	668	N/A	39%	N/A

only_wildebeest	3,057	3,057*	1,756	0	0	1,301	N/A	57%	N/A
only_other	6,183	6,183*	3,992	0	0	2,191	N/A	65%	N/A
multi_species	491	25	10	282	184	15	5%	40%	9%
multi_including_ wildebeest	323	319*	187	2	2	132	99%	59%	74%
multi_including_ none_of_the_five	124	124*	77	0	0	47	N/A	62%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets which do not contain the desired speciesòabove.

The tests using the nighttime test sets were not available for this test, as there were no images containing warthogs in this test set.

Results: Wildebeest

The results for **animal_type_wildebeest** were as follows.

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	3,380	1,399	39,569	7,321	1,981	16%	41%	23%
daytime	368	25	8	281	62	17	11%	32%	17%
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	32,306	0	0	4,428	N/A	88%	N/A
non_blank	13,536	3,380	1,399	7,263	2,893	1,981	33%	41%	36%
one_animal	6,172	602	198	4,108	1,462	404	12%	33%	18%
simple	4,583	1,061	488	2,409	1,113	573	30%	46%	37%
complex	2,781	1,717	713	746	318	1,004	69%	42%	52%
single_species	13,045	3,057	1,261	7,152	2,836	1,796	31%	41%	35%
only_buffalo	1,169	1,169*	817	0	0	352	N/A	70%	N/A
only_elephant	1,190	1,190*	890	0	0	300	N/A	75%	N/A
only_ostrich	356	356*	316	0	0	40	N/A	89%	N/A
only_warthog	1,090	1,090*	868	0	0	222	N/A	80%	N/A
only_wildebeest	3,057	3,057	1,261	0	0	1,796	N/A	41%	N/A
only_other	6,183	6,183*	4,261	0	0	1,922	N/A	69%	N/A
multi_species	491	323	138	111	57	185	71%	43%	53%

multi_including_ wildebeest	323	323	138	0	0	185	N/A	43%	N/A
multi_including_ none_of_the_five	124	124*	83	0	0	41	N/A	67%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets which do not contain the desired speciesòabove.

The tests using the nighttime test sets were not available for this test, as there were no images containing wildebeest in this test set.

Results: Summary

The results for **entire_dataset**, across all five tests were as follows:

Test	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
animal_type_ buffalo	50,270	1,187	387	41,179	7,904	800	5%	33%	8%
animal_type_ elephant	50,270	1,197	277	44,691	4,382	920	6%	23%	9%
animal_type_ ostrich	50,270	357	219	29,181	20,732	138	1%	61%	2%
animal_type_ warthog	50,270	1,115	432	35,643	13,512	683	3%	39%	6%
animal_type_ wildebeest	50,270	3,380	1,399	39,569	7,321	1,981	16%	41%	23%
total	50,270	7,236	2,714	190,263	53,851	4,522	5%	38%	9%
average per species							6%	39%	10%

Overall

Overall, the precision was low - 6% or less. This means that the platform has a tendency to over-classify, and an extremely high chance (94% or more) of being incorrect (i.e. the species is not present) when it reports the presence of one of the five target species in a given image. As such, the platform is not yet useful as a means to pre-determine which species is present. We recognise that emptiness detection was the primary focus of the detectors at this stage, so this is not unexpected.

Overall, precision for wildebeest was about 10% higher. Still low, but considerably better than the other species. Whether this is in part due to the high prevalence of wildebeest in Serengeti images is unclear.
Recall - the ability of the platform to correctly recognize the presence of a species when it is present and not miss any - was generally better - varying between 23% (elephants) and 61% (ostriches). This level of performance is moderate, and in the case of ostriches, quite good. However we have observed that the platform identifies the presence of ostriches in the vast majority of images, so this, combined with the low number of ostriches in our dataset, may simply be a lucky ñscattergunò success.

Day vs Night

None of the target species were present in our night time test set, therefore we are unable to say whether daylight makes species identification easier.

Simple vs Complex

There was in general a better precision and recall for species identification in complex images (those with more than 5 animals) than in simple ones (5 animals or fewer). However, buffalo gave identical recall for both simple and complex images.
Also worthy of note is the fact that for wildebeest, that precision (likelihood of being correct about the species being present) was much higher, and quite good, in complex images - 69% - than in simple images, where precision was poor - 30%. This means the wildebeest detector is more likely to be correct when it does report a buffalo - but, similarly to ostriches, this is probably explained by the scattergun theory - in this case because there *are* a very high number of buffalo in Serengeti images, therefore the numerous reports of buffalo in images are on average more likely to be correct.

Species Mix

In general precision and recall were higher for multi species images than single species images. This is again probably because multiple detectors contributed to the decision in those cases. Also, given that this is less true for ostriches, and more true for wildebeest, it is again an unreliable conclusion that is likely influenced by the prevalence of each species in the test data.

Specific Species

As mentioned above in ñOverallò, precision was generally very poor - though slightly better for those species that are more commonly found. Recall was moderate, and heavily improved by the rarity of a given species. In short, we should be careful about drawing conclusions about the accuracy for specific species in these unequal test sets.

TP-202-03 - Test animal counting across a season of subjects

For this test, we wanted to evaluate how the MICO platform performs at detecting the correct number of animals. This corresponds to the Accuracy Definition for ñAnimal Countò under Methodology and Planning in the Test Plan.

Metrics for test sets with a mix of animal counts

For testing the correct detection of a specific animal count, we defined ten different tests, **animal_count_1** to **animal_count_10**, each corresponding to a trying to detect a specific number of animals. In each test, we did not have a tolerance threshold. In other words, when testing for an animal count of 6 (**animal_count_6**) then any number other than 6 is counted as a fail, even 5 or 7.

For a given test **animal_count_x**, the metrics are as follows

- Relevance is defined as ñThe image contains exactly x animalsò
- Selected is defined as ñThe image is identified as containing exactly x animalsò(i.e. the detector detects x regions)
- This leads to the following definitions:
  - True positive: Image with x animals correctly identified as having x animals
  - True negative: Image with a number of animals other than x correctly identified as having a number of animals other than x.
  - False negative: Image with x animals incorrectly identified as having a number of animals other than x.
  - False positive: Image with a number of animals other than x incorrectly identified as having x animals
  - Precision: The percentage of images identified as having x animals which do indeed contain x animals.
  - Recall: The percentage of images which actually contain x animals that are correctly identified as containing x animals.
  - F1 score: overall measure of this testôs accuracy, combining precision and recall

Metrics for test sets where the desired count is impossible

A number of the test sets, by definition, do not contain the tested for number of animals. (For example, blank or multi_species cannot contain 1 animal. Therefore for these tests, rather than omitting the test, we looked at how good the detector was at detecting that the number of animals was *not* equal to the test value. In other words, we used an alternate definition of relevance in these tests, for a given test **animal_count_x**:

- Relevance is defined as ñThe image contains a number of animals other than xò
- Selected is defined as ñThe image is identified as containing a number of animals other than xò
- This leads to the following definitions:
  - True positive: Image with some number of animals other than x correctly identified as having some number of animals other than x
  - True negative: Not applicable in this case, as there are no images containing x animals.
  - False negative: Image with some number of animals other than x incorrectly identified as having a x animals.

- ð False positive: Not applicable in this case, as there are no images containing x animals.
- ð Precision: Not applicable in this case, as there are no non-relevant images.
- ð Recall: The percentage of images which do not contain x animals that are correctly identified as containing some other number of animals than x.
- ð F1 score: Not applicable in this case, as there is no figure for precision.

Test sets where this alternate set of metrics was used are marked with an asterisk in the results table below.

Results: 1 animal

The results for **animal_count_1** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	6,172	1,188	34,776	9,322	4,984	11%	19%	14%
daytime	368	34	10	272	62	24	14%	29%	19%
nighttime	183	105	13	65	13	92	50%	12%	20%
blank	36,734	36,734*	28,727	0	0	8,007	N/A	78%	N/A
non_blank	13,536	6,172	1,188	6,049	1,315	4,984	47%	19%	27%
one_animal	6,172	6,172	1,188	0	0	4,984	N/A	19%	N/A
simple	4,583	4,583*	3,756	0	0	827	N/A	82%	N/A
complex	2,781	2,781*	2,293	0	0	488	N/A	82%	N/A
single_species	13,045	6,172	1,188	5,642	1,231	4,984	49%	19%	28%
only_buffalo	1,169	464	101	561	144	363	41%	22%	28%
only_elephant	1,190	610	146	466	114	464	56%	24%	34%
only_ostrich	356	287	76	54	15	211	84%	26%	40%
only_warthog	1,090	836	160	202	52	676	75%	19%	31%
only_wildebeest	3,057	602	108	2,054	401	494	21%	18%	19%
only_other	6,183	3,373	597	2,305	505	2,776	54%	18%	27%
multi_species	491	491*	407	0	0	84	N/A	84%	N/A
multi_including_ wildebeest	323	323*	272	0	0	51	N/A	81%	N/A
multi_including_ none_of_the_five	124	124*	100	0	0	24	N/A	84%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

Results: 2 animals

The results for **animal_count_2** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	2,109	285	41,942	6,219	1,824	4%	14%	7%
daytime	368	12	1	295	61	11	2%	8%	3%
nighttime	183	9	0	158	16	9	0%	0%	N/A
blank	36,734	36,734*	32,183	0	0	4,551	N/A	88%	N/A
non_blank	13,536	2,109	285	9,759	1,668	1,824	15%	14%	14%
one_animal	6,172	6,172*	5,284	0	0	888	N/A	86%	N/A
simple	4,583	2,109	285	2,127	347	1,824	45%	14%	21%
complex	2,781	2,781*	2,348	0	0	433	N/A	84%	N/A
single_species	13,045	2,031	279	9,403	1,611	1,752	15%	14%	14%
only_buffalo	1,169	195	26	821	153	169	15%	13%	14%
only_elephant	1,190	210	33	854	126	177	21%	16%	18%
only_ostrich	356	42	6	252	62	36	9%	14%	11%
only_warthog	1,090	162	16	789	139	146	10%	10%	10%
only_wildebeest	3,057	333	48	2,296	428	285	10%	14%	12%
only_other	6,183	1,089	150	4,391	703	939	18%	14%	15%
multi_species	491	78	6	356	57	72	10%	8%	9%
multi_including_ wildebeest	323	12	1	275	36	11	3%	8%	4%
multi_including_ none_of_the_five	124	49	2	57	18	47	10%	4%	6%

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

Results: 3 animals

The results for **animal_count_3** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	1,232	124	45,090	3,948	1,108	3%	10%	5%

daytime	368	11	1	318	39	10	3%	9%	4%
nighttime	183	2	0	174	7	2	0%	0%	N/A
blank	36,734	36,734*	34,041	0	0	2,693	N/A	93%	N/A
non_blank	13,536	1,232	124	11,049	1,255	1,108	9%	10%	9%
one_animal	6,172	6,172*	5,552	0	0	620	N/A	90%	N/A
simple	4,583	1,232	124	3,007	344	1,108	26%	10%	15%
complex	2,781	2,781*	2,490	0	0	291	N/A	90%	N/A
single_species	13,045	1,190	116	10,648	1,207	1,074	9%	10%	9%
only_buffalo	1,169	108	6	940	121	102	5%	6%	5%
only_elephant	1,190	125	16	951	114	109	12%	13%	13%
only_ostrich	356	15	0	305	36	15	0%	0%	N/A
only_warthog	1,090	69	4	923	98	65	4%	6%	5%
only_wildebeest	3,057	303	31	2,476	278	272	10%	10%	10%
only_other	6,183	570	59	5,053	560	511	10%	10%	10%
multi_species	491	42	8	401	48	34	14%	19%	16%
multi_including_ wildebeest	323	20	4	277	26	16	13%	20%	16%
multi_including_ none_of_the_five	124	22	4	86	16	18	20%	18%	19%

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleò above.

Results: 4 animals

The results for **animal_count_4** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	720	72	46,928	2,622	648	3%	10%	4%
daytime	368	7	0	338	23	7	0%	0%	N/A
nighttime	183	2	0	170	11	2	0%	0%	N/A
blank	36,734	36,734*	35,124	0	0	1,610	N/A	96%	N/A
non_blank	13,536	720	72	11,804	1,012	648	7%	10%	8%
one_animal	6,172	6,172*	5,711	0	0	461	N/A	93%	N/A
simple	4,583	720	72	3,533	330	648	18%	10%	13%

complex	2,781	2,781*	2,560	0	0	221	N/A	92%	N/A
single_species	13,045	674	68	11,386	985	606	6%	10%	8%
only_buffalo	1,169	66	4	1,011	92	62	4%	6%	5%
only_elephant	1,190	76	8	1,026	88	68	8%	11%	9%
only_ostrich	356	3	2	334	19	1	10%	67%	17%
only_warthog	1,090	13	1	1,005	72	12	1%	8%	2%
only_wildebeest	3,057	178	18	2,623	256	160	7%	10%	8%
only_other	6,183	338	35	5,387	458	303	7%	10%	8%
multi_species	491	46	4	418	27	42	13%	9%	10%
multi_including_ wildebeest	323	26	3	275	22	23	12%	12%	12%
multi_including_ none_of_the_five	124	16	1	105	3	15	25%	6%	10%

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

Results: 5 animals

The results for **animal_count_5** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	522	36	47,859	1,889	486	2%	7%	3%
daytime	368	8	0	347	13	8	0%	0%	N/A
nighttime	183	1	0	173	9	1	0%	0%	N/A
blank	36,734	36,734*	35,660	0	0	1,074	N/A	97%	N/A
non_blank	13,536	522	36	12,199	815	486	4%	7%	5%
one_animal	6,172	6,172*	5,826	0	0	346	N/A	94%	N/A
simple	4,583	522	36	3,769	292	486	11%	7%	8%
complex	2,781	2,781*	2,604	0	0	177	N/A	94%	N/A
single_species	13,045	496	35	11,767	782	461	4%	7%	5%
only_buffalo	1,169	36	4	1,064	69	32	5%	11%	7%
only_elephant	1,190	58	5	1,081	51	53	9%	9%	9%
only_ostrich	356	6	2	332	18	4	10%	33%	15%
only_warthog	1,090	3	0	1,048	39	3	0%	0%	N/A

only_wildebeest	3,057	172	11	2,666	219	161	5%	6%	5%
only_other	6,183	221	13	5,576	386	208	3%	6%	4%
multi_species	491	26	1	432	33	25	3%	4%	3%
multi_including_ wildebeest	323	17	1	282	24	16	4%	6%	5%
multi_including_ none_of_the_five	124	8	0	107	9	8	0%	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

Results: 6 animals

The results for **animal_count_6** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	400	22	48,524	1,346	378	2%	6%	2%
daytime	368	2	0	353	13	2	0%	0%	N/A
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	36,010	0	0	724	N/A	98%	N/A
non_blank	13,536	400	22	12,514	622	378	3%	6%	4%
one_animal	6,172	6,172*	5,904	0	0	268	N/A	96%	N/A
simple	4,583	4,583*	4,347	0	0	236	N/A	95%	N/A
complex	2,781	400	22	2,263	118	378	16%	6%	8%
single_species	13,045	362	19	12,093	590	343	3%	5%	4%
only_buffalo	1,169	37	1	1,078	54	36	2%	3%	2%
only_elephant	1,190	37	1	1,090	63	36	2%	3%	2%
only_ostrich	356	3	0	344	9	3	0%	0%	N/A
only_warthog	1,090	7	1	1,041	42	6	2%	14%	4%
only_wildebeest	3,057	144	11	2,764	149	133	7%	8%	7%
only_other	6,183	134	5	5,776	273	129	2%	4%	2%
multi_species	491	38	3	421	32	35	9%	8%	8%
multi_including_ wildebeest	323	18	3	281	24	15	11%	17%	13%
multi_including_ none_of_the_five	124	14	0	103	7	14	0%	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

The test using the nighttime test set was not available for this test, as there were no nighttime images containing 6 animals.

Results: 7 animals

The results for **animal_count_7** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	307	18	48,906	1,057	289	2%	6%	3%
daytime	368	3	0	359	6	3	0%	0%	N/A
nighttime	183	2	0	172	9	2	0%	0%	N/A
blank	36,734	36,734*	36,188	0	0	546	N/A	99%	N/A
non_blank	13,536	307	18	12,718	511	289	3%	6%	4%
one_animal	6,172	6,172*	5,965	0	0	207	N/A	97%	N/A
simple	4,583	4,583*	4,389	0	0	194	N/A	96%	N/A
complex	2,781	307	18	2,364	110	289	14%	6%	8%
single_species	13,045	277	13	12,276	492	264	3%	5%	3%
only_buffalo	1,169	41	1	1,087	41	40	2%	2%	2%
only_elephant	1,190	31	0	1,124	35	31	0%	0%	N/A
~~only_ostrich~~	356	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
~~only_warthog~~	1,090	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
only_wildebeest	3,057	98	11	2,807	152	87	7%	11%	8%
only_other	6,183	107	1	5,850	226	106	0%	1%	1%
multi_species	491	30	5	442	19	25	21%	17%	19%
multi_including_ wildebeest	323	26	5	287	10	21	33%	19%	24%
multi_including_ none_of_the_five	124	3	0	115	6	3	0%	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

The tests using the only_ostrich and only_warthog test sets were not available for this test, as there were no images containing 7 animals in these test sets.

Results: 8 animals

The results for **animal_count_8** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	245	2	49,209	816	243	0%	1%	0%
daytime	368	1	0	365	2	1	0%	0%	N/A
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	36,354	0	0	380	N/A	99%	N/A
non_blank	13,536	245	2	12,855	436	243	0%	1%	1%
one_animal	6,172	6,172*	6,007	0	0	165	N/A	97%	N/A
simple	4,583	4,583*	4,408	0	0	175	N/A	96%	N/A
complex	2,781	245	2	2,440	96	243	2%	1%	1%
single_species	13,045	225	1	12,405	415	224	0%	0%	0%
only_buffalo	1,169	28	0	1,119	22	28	0%	0%	N/A
only_elephant	1,190	13	0	1,157	20	13	0%	0%	N/A
~~only_ostrich~~	356	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
~~only_warthog~~	1,090	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
only_wildebeest	3,057	103	1	2,820	134	102	1%	1%	1%
only_other	6,183	81	0	5,881	221	81	0%	0%	N/A
multi_species	491	20	1	450	21	19	5%	5%	5%
multi_including_ wildebeest	323	17	1	289	17	16	6%	6%	6%
multi_including_ none_of_the_five	124	1	0	120	3	1	0%	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

The tests using the nighttime, only_ostrich and only_warthog test sets were not available for this test, as there were no images containing 8 animals in these test sets.

Results: 9 animals

The results for **animal_count_9** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	158	5	49,514	598	153	1%	3%	1%
~~daytime~~	368	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734*	36,460	0	0	274	N/A	99%	N/A
non_blank	13,536	158	5	13,054	324	153	2%	3%	2%
one_animal	6,172	6,172*	6,044	0	0	128	N/A	98%	N/A
simple	4,583	4,583*	4,454	0	0	129	N/A	97%	N/A
complex	2,781	158	5	2,556	67	153	7%	3%	4%
single_species	13,045	138	4	12,597	310	134	1%	3%	2%
only_buffalo	1,169	20	1	1,129	20	19	5%	5%	5%
only_elephant	1,190	17	0	1,147	26	17	0%	0%	N/A
~~only_ostrich~~	356	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
~~only_warthog~~	1,090	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
only_wildebeest	3,057	51	2	2,918	88	49	2%	4%	3%
only_other	6,183	50	1	5,975	158	49	1%	2%	1%
multi_species	491	20	1	457	14	19	7%	5%	6%
multi_including_ wildebeest	323	12	1	301	10	11	9%	8%	9%
multi_including_ none_of_the_five	124	2	0	120	2	2	0%	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

The tests using the daytime, nighttime, only_ostrich and only_warthog test sets were not available for this test, as there were no images containing 9 animals in these test sets.

Results: 10 animals

The results for **animal_count_10** were as follows:

Test Set used	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
entire_dataset	50,270	165	3	49,664	441	162	1%	2%	1%
daytime	368	1	0	365	2	1	0%	0%	N/A
~~nighttime~~	183	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
blank	36,734	36,734	36,544	0	0	190	N/A	99%	N/A
non_blank	13,536	165	3	13,120	251	162	1%	2%	1%
one_animal	6,172	6,172	6,063	0	0	109	N/A	98%	N/A
simple	4,583	4,583	4,489	0	0	94	N/A	98%	N/A
complex	2,781	165	3	2,568	48	162	6%	2%	3%
single_species	13,045	153	3	12,651	241	150	1%	2%	2%
only_buffalo	1,169	32	0	1,118	19	32	0%	0%	N/A
only_elephant	1,190	7	1	1,169	14	6	7%	14%	9%
~~only_ostrich~~	356	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
~~only_warthog~~	1,090	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
only_wildebeest	3,057	71	1	2,928	58	70	2%	1%	2%
only_other	6,183	43	1	6,003	137	42	1%	2%	1%
multi_species	491	12	0	469	10	12	0%	0%	N/A
multi_including_ wildebeest	323	10	0	303	10	10	0%	0%	N/A
multi_including_ none_of_the_five	124	2	0	122	0	2	N/A	0%	N/A

* : For these tests, the alternate definition of relevance was used, as described in ñMetrics for test sets where the desired count is impossibleòabove.

The tests using the nighttime, only_ostrich and only_warthog test sets were not available for this test, as there were no images containing 10 animals in these test sets.

Results: Summary

The results for **entire_dataset**, across all ten tests were as follows:

Test	*Sample Size*	*Relevant Images*	*True Pos*	*True Neg*	*False Pos*	*False Neg*	*Precision*	*Recall*	*F1 Score*
animal_count_1	50,270	6,172	1,188	34,776	9,322	4,984	11%	19%	14%
animal_count_2	50,270	2,109	285	41,942	6,219	1,824	4%	14%	7%
animal_count_3	50,270	1,232	124	45,090	3,948	1,108	3%	10%	5%
animal_count_4	50,270	720	72	46,928	2,622	648	3%	10%	4%
animal_count_5	50,270	522	36	47,859	1,889	486	2%	7%	3%
animal_count_6	50,270	400	22	48,524	1,346	378	2%	6%	2%
animal_count_7	50,270	307	18	48,906	1,057	289	2%	6%	3%
animal_count_8	50,270	245	2	49,209	816	243	0%	1%	0%
animal_count_9	50,270	158	5	49,514	598	153	1%	3%	1%
animal_count_10	50,270	165	3	49,664	441	162	1%	2%	1%
total	50,270	12,030	1,755	462,412	28,258	10,275	6%	15%	8%
average per count							3%	8%	4%

Overall

Both precision and recall were uniformly very low. The extractor is very poor at estimating the number of animals when it does detect animals, and very likely to detect an incorrect number of animals.

The only good results were obtained in those cases where the relevance definition was switched; the extractor is extremely good at saying when the number animals present is **not** equal to a target - however, this is unremarkable, as it is very easy for any random guess at the number of animals to be wrong.

For this reason, these reversed tests under animal_count_x should probably be discarded as not useful. However at this stage they are left in the report for completeness.

Day vs Night

For those values of x where MICO was able to get the correct number of animals at all - 1, 2 and 3, it did seem that daytime images achieved a higher recall than nighttime ones, almost twice as much in some cases. In other words, MICO is better at being right on the number of animals, regardless of target, in the daytime. Performance is better for smaller values of n - especially 1, but even then, at recall of 29%, the recall is still poor.

The same effect seems to hold for precision at low values of x too, with a better precision during daytime than at night for 2 or 3 animals.

Interestingly, for 1 animal, the opposite seems to hold true, with a precision of 50% at night compared to only 14% in the day. One theory for this is that fewer ñdistracting featuresò such as clouds, bushes or shadows are available at night to be mistaken for animals, thus the false positives are low. It is unclear why this effect would not continue for precision with all values of x at night, however the number of relevant images for each value of x is so low that this could be a fluke. Further testing with a larger test set would be desirable.

For 4 animals or more, precision and recall were both 0 for the day/night test sets so no distinction can be made between night and day.

Simple vs Complex

As mentioned above under ñOverallò, the number of animals present doesnôt make any noticeable difference to precision or recall. The scores remain almost identical for all x values - always below 10% and usually around 3%. Where the relevance definition was inverted, the precisions and recalls are also inverted (90% and 97%). Again, this is unremarkable and these reversed tests should probably be discarded.

Species Mix

The number of different species in an image did not make much difference in the ability of MICO to count the number of animals correctly. Images with more than one species were a generally more likely to be counted correctly than those with one single species, by a factor of 3-10%, though there are exceptions to this. This is probably because the results of multiple different species detectors contributed to the answer.

Specific Species

MICOôs ability to correctly count the number of animals correctly did vary quite a lot between the 5 species, however the precision and recall were poor (10-15% or less) for almost all values of x for all 5 species. Wildebeest in general performed slightly better for both metrics, though this may be influenced by the very high number of wildebeest in Serengeti photos compared to other species (it is the most prevalent).

One exception is when determining whether an image contains exactly 1 animal - the extractors for warthogs and ostriches performed exceptionally well for this - with precision values of 75% and 84% respectively. It is unclear why these particular species with this particular x value would be better than others. Also for 1 animal the precision for wildebeest is especially low - 21% - meaning MICO is more likely to miscount the number of wildebeest in solo wildebeest pictures than for other species. Again the reason for this is unclear.

Overall Analysis of Platform for TP-202 (Animal Detection)

Latency

We measured the time it took from sending an image to the platform, until we had a final result back.

Extractor	Sample size	Mean	Standard dev.	90th percentile
Y2: DPM	1397	5.43	9.75	5.90
YOLO	9870	383.34	17.83	597.32

For the YOLO extractor, we sent and timed a total of 9870 images. We found a mean time of 383 seconds, with a standard deviation of 17.83 seconds.

Throughput

While the YOLO extractor has much better precision and recall, it is also immensely slower than the previous DPM extractor. It is suspected that the measured slowness is due to a bug in the system at time of validation, since the numbers were about 10 times higher than expected. Additionally, by moving the processing to a discrete GPU, speed is expected to increase by at least another order of magnitude, and probably more.

The test server did not have any graphics card, and could not quickly be equipped with one for the validation. Since with the tested speed a small dataset would take over a year of processing time, some additional debugging and validation might be needed in order to evaluate how cost-effective the YOLO extractor is.

Stability

In year 3 of the MICO project, a lot of effort went into a newer version of the overarching platform. In the previous (year 2) validation report we had a lot of stability issues, but for the final validation the platform as of version 3 has been solid. We only experienced a single case of downtime, which came as a result of log files filling up the entire disk, which is easily remedied with some log rotation rules.

Modularity

The year 2 version of the platform had an issue where it could only run one pipeline at a time, meaning it couldn't run images and text pipelines simultaneously. We're happy to report that this issue is completely resolved with the release of platform version 3.

Integration

We found the platform really easy to integrate. Integration for the demo application was very easy and done in about 100 lines of Ruby code. (See https://github.com/zooniverse/mico-api-client).

Usefulness

The ability to get an 84% precise opinion (with the DPM extractor) upon whether an image is blank is **immediately useful** to Zooniverse. Our data scientist ran a simulation of how many classifications would be needed to correctly classify all of the 300,000 images in Season 8 / TD-12 as blank or non-blank, and found that:

- Normally, without MICO, we would require at least 3 people to agree an image is blank before we can safely retire it and define it as blank (with a tiny error rate of only 0.18%)
- Adding MICOôs determination into the mix, we can drop this down to just 2 users required where MICO thinks the image is blank (when MICO says it is non-blank, we still need 3 users).
- Without MICO, and using 3 people per image, we need 817,015 classifications to correctly classify all blank and non-blank images in TD-12.
- With MICO used as described above, we need only 702,404 classifications:
  - **Using the emptiness detection, MICO can save us over 115,000 classifications and allow us complete a season of data on Snapshot Serengeti in Ö of the time - which is of the order of several days to a week of volunteer time.**

The YOLO extractor is much better still, which would result in even better savings. However, it needs some more testing to see what type of hardware is needed to speed up the detection. Weôve been told especially a graphics card would make a big difference, and the server that the Zooniverse test instance was running on wasnôt equipped with any discrete graphics chip at all.

Caveat: These figures are estimates, based on initial calculations and simulations, and have not been tested ñin the wildò

Nonetheless, **this is an excellent and exciting prospect for Zooniverse and a clear positive outcome for the MICO project.**

Evaluation of Testing Approach for TE-202 (Animal detection)

In general, our testing strategy has been useful for giving an overview of the platformôs ability, especially when looking at the accuracy across the entire dataset. However, some weaknesses were identified:

- The large variability in the chance that a given image will contain a certain species makes it much harder to trust the precision and recall values for detection of specific species. In future testing we will need to do more to normalize the number of each species present in all test data sets. For example, the high occurrence of wildebeest and the low number of ostriches have clearly biased those detectorsôresults for precision and recall respectively.
- The low number of images for day and night time sets has limited the usefulness of those tests. In future we hope to design a solution to properly determine the day-or-night status of all images in our test set, this will mean fewer tests having to be discarded due to not having enough occurrences of a particular number of animals or species in the day/night test sets.

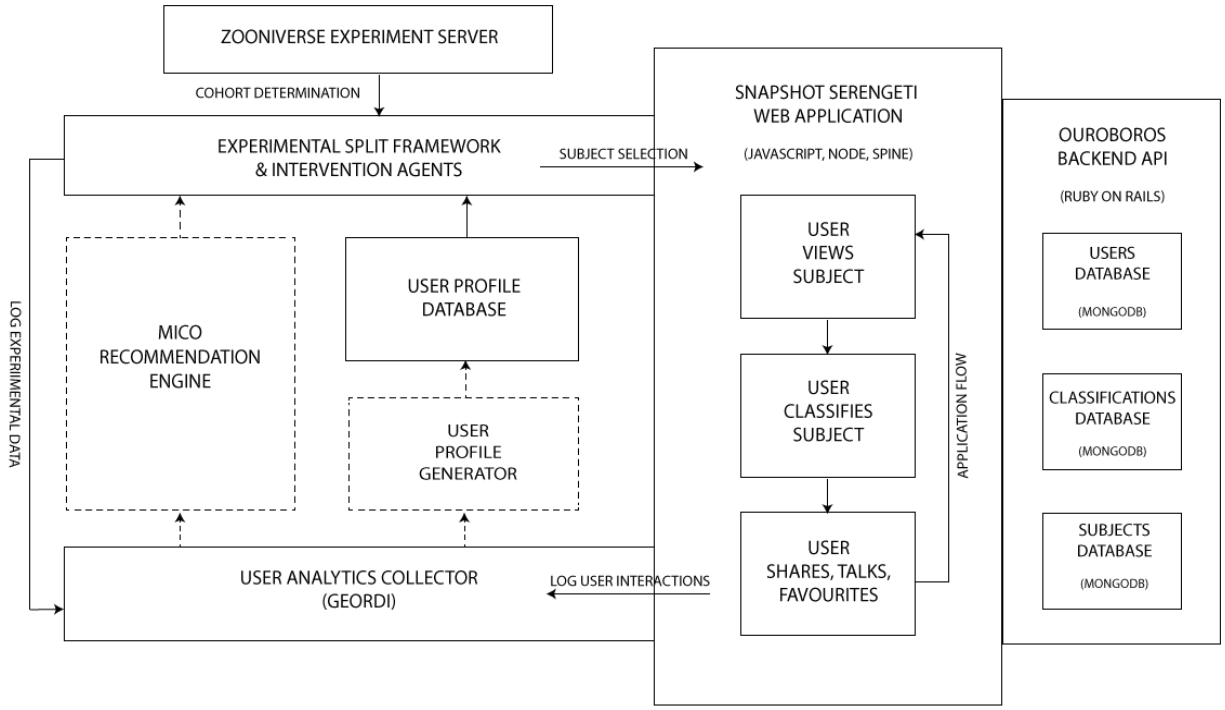
- Similarly, having more processing time available on the platform will allow us to create much larger test sets for the 5 specific species. This too will increase the accuracy of species-specific tests. We may need to look for additional ostrich images in past seasons of Serengeti data outside of TD-12, since there are so few present in even the full 300,000 images.
- The decision to invert the relevancy definition in some cases to open up more possible test sets had some limited success in telling us more about the platformôs capability, but this approach needs more refinement as some of these inverted rows are misleading and show a high performance quality when that isnôt really the case (e.g. confirming that an image doesnôt have exactly 3 animals).

In future test cycles we hope to address many of these weaknesses.

Background and Context of TE-506: Cross modal content recommender

Some progress was made by Zaizi towards a Prediction-IO based machine learning solution for subject recommendation. Zooniverse made parallel efforts to build a simple subject recommendation system which suited our existing user base, using a flexible combination of MySQL queries and Python scripts, which would serve the same purpose.

This is marked as a dotted box called ñUser Profile Generatoròin the architecture diagram shown below (which was included in the linked First Prototype document):



The original Test Plan for this work was written at a time when it was still expected that the Zaizi solution would become part of the MICO platform, so some of it is no longer relevant. However,

the Zooniverse user profile generator/recommender system was found to be perfectly adequate for the needs of the Happy User Experiment.

What follows is an account of what was done to develop and functionally test each stage of the Zooniverse recommender system (ñuser profile generatorò), which meets the functional capabilities for TE-506 of being able to recommend subjects for a user based on their behaviour and inferred preferences.

We are confident that the work done is in line with the goals of WP5, and that it made more sense to keep the current test plan rather than abandon it and re-plan from scratch.

We concluded based on the results of TP-506-04 below that the theory of using user personalised subject selection based on ñpreferredò species was not particularly useful (given that the use of the recommended subjects damaged rather than improved participation), and the focus of MICO/Zooniverse work moved towards the use of animal detection and sentiment analysis/NER as cross media sources for recommendations, while the research angle moved towards investigation of the impact of blank subjects.

TP-506-01 - Test that species preferences match user behaviour

This test focussed on validating that a recommendation system trained upon user analytics data and species content data could correctly identify a userôs species preferences.

First, subject species content (TD-16) was calculated for all past Snapshot Serengeti users using a combination of MySQL queries and Python scripts [ñDETERMINE SUBJECT CONTENTòabove]

DETERMINE SUBJECT CONTENT - Pseudocode

```
for each subject
  combine all user classifications for this subject
  find the mode species (ie. that which the most users classified)
  store this as the dominant species for this subject
next subject
```

This was done by aggregating all user classifications for the subjects and taking the most commonly occurring answer, or ñblankò, as the correct indication of subject content - as shown in the pseudocode above. Where more than one species was present (only 6% of images) we take the more numerous species.

Sample of TD-16 data

id	subjectID	species
335949	ASG0000017	blank
335950	ASG0000018	blank
335951	ASG0000019	elephant
335952	ASG000001a	elephant
335953	ASG000001b	leopard
335954	ASG000001c	blank
335955	ASG000001d	giraffe
335956	ASG000001e	blank
335957	ASG000001f	blank
335958	ASG000001g	blank
335959	ASG000001h	dikdik
335960	ASG000001i	blank
335961	ASG000001j	blank
335962	ASG000001k	gazellethomsons
335963	ASG000001l	blank
335964	ASG000001m	impala

Now that we have this data, we are then able to determine userôs preferred species by looking at which subjects they prefer.
The general approach for generating TD-14 was as follows (again this was done using a combination of MySQL queries and Python scripts):

- ð Identify and tally indications of positive interest in a subject from the analytics data [ñCOLLECT SUBJECT INTEREST DATAòin diagram above]. We included the following user events:
  - ȝ User favourites the subject
  - ȝ User shares the subject via Facebook, pinterest or Twitter
  - ȝ User views the discussion page for a subject
  - ȝ User views the map for a subject

COLLECT SUBJECT INTEREST DATA - Pseudocode

```
for each user
  for each “positive interest” user analytics event
    increment interest counter for that subject for that user
  next analytics event
  store total interest by subject for each user
next user
```

- ð Convert each tallied indication of interest in a subject, to an indication of interest in the dominant species (or ñblankò) of that subject [ñDETERMINE SPECIES INTERESTòin diagram above]. Each indication of interest is assigned a simple numerical score of 1.

DETERMINE SPECIES INTEREST - Pseudocode

```
for each user
  for each subject with positive interest
    determine which species is present (from TD-16)
    increment interest counter for that species for that user
  next subject
  store total "expressions of interest" per species for this user
next user
```

- Assign a score per user, per species, based on the sum of total expressions of interest in that species by that user [ñGENERATE USER PROFILESòin diagram above]. So for example, if a user favourites an image of a zebra and shares two images containing lions, they would score 2.0 for lions and 1.0 for zebra. A higher score indicates a stronger interest in that species, and the two highest scoring species for a user can be deemed as that userôs ñfavoriteòspecies.

GENERATE USER PROFILES - Pseudocode

```
for each user
  for each species with positive interest
    add an entry in that user's profile that they like that species
    count number of expressions of interest for that species
    assign that number as a score to the user profile entry
  next species
next user
```

Sample of data from TD-14 user profile data

id	userID	score	scoreType	secondaryID
17611	998037	3.000	interestInSpecies	elephant
17609	997519	1.000	interestInSpecies	zebra
17610	997519	1.000	interestInSpecies	wildebeest
17605	993649	1.000	interestInSpecies	elephant
17606	993649	1.000	interestInSpecies	baboon
17607	993649	1.000	interestInSpecies	blank
17608	993649	2.000	interestInSpecies	zebra
17603	99354	1.000	interestInSpecies	wildebeest
17604	99354	2.000	interestInSpecies	blank
17597	988918	1.000	interestInSpecies	wildebeest
17598	988918	2.000	interestInSpecies	zebra
17599	988918	1.000	interestInSpecies	blank
17600	988918	1.000	interestInSpecies	gazellegrants
17601	988918	1.000	interestInSpecies	gazellethomsons
17602	988918	1.000	interestInSpecies	human
17596	988799	1.000	interestInSpecies	zebra
17595	986174	1.000	interestInSpecies	wildebeest

Testing approach

Since the recommendation system built here was a series of scripts and queries rather than a service, testing focussed on data quality - ensuring the data was correct and accurately representing userôs species preferences. This was done as follows: We randomly selected a few users and inspected their user profile data (TD-14) to identify their two favourite species (those with the highest score for that user). We then randomly selected 5 subjects which those users had expressed positive interest in and confirmed that the subjects contained predominantly those two favorite species. This test was successful. This approach is however self-referential to the dataset, and no statistics were calculated on how strong the correlation between our chosen expressions of interest and actually liking that animal is.

The KPIs planned (precision, recall, F1-measure) were not tested since we are no longer in a comparison to external known good data scenario.

The ñchanged behaviourò borderline case was tested by modifying user subject positive interest tallies to favour subjects containing different species, and re-generating the user profile data accordingly, and userôs favourite species were observed to have changed correctly.

TP-506-02 - Test that the subjects recommended match the preferred species

This test focussed on ensuring that the correct subjects would be recommended (in line with the calculated species preferences for each user) for a user once that userôs species preferences were known.

Subject recommendation was performed as follows, using a combination of MySQL queries and Python scripts:

- A random set of 100 subjects was chosen for each species and ñblankò, using the TD-16 data. We ensured that these images did not contain other secondary species [ñGENERATE SUBJECT SETS PER SPECIESò in diagram above].

GENERATE USER PROFILES - Pseudocode

```
for each species
  initialize subject set for this species
  while subject set contains < 100 items
    current subject = random subject
    while current subject does not contain this species
      pick another random subject
    end while
    add this subject to the subject set
  end while
next species
```

Sample of ñSUBJECT SETS PER SPECIESò data

id	species	subjectIDs
139	gazelleGrants	ASG000dua0,ASG0011llk,ASG0002y3l,ASG0002hbw,ASG0002kdt,ASG0000283,ASG000…
140	gazelleThomsons	ASG000q46r,ASG0008l4l,ASG0002lg7,ASG0008lgm,ASG0002qyn,ASG0002rvr,ASG000…
141	genet	ASG00022xy,ASG000rtjl,ASG0000aho,ASG00016ly,ASG000ofkd,ASG0004bn9,ASG0008…
142	giraffe	ASG0002psj,ASG000bo9d,ASG000qin5,ASG0005qsl,ASG000qlj5,ASG000qpdh,ASG000…
143	guineaFowl	ASG0011ghv,ASG000qel5,ASG0003v8j,ASG0003ve5,ASG00001ap,ASG0006023,ASG00…
144	hare	ASG00070d8,ASG0000fpz,ASG0002ujq,ASG0004bvh,ASG000n0c9,ASG0000fqc,ASG00…
145	hartebeest	ASG0003xc2,ASG0003wxl,ASG0003xs1,ASG0003vvc,ASG0003vy0,ASG0003vu4,ASG00…
146	hippopotamus	ASG0002ujo,ASG0002gmz,ASG000rsqt,ASG000d45q,ASG000bd20,ASG000oigx,ASG00…
147	honeyBadger	ASG0002dip,ASG00038tl,ASG000397z,ASG000s8vx,ASG0011rxb,ASG000ackq,ASG000…
148	human	ASG000bkfb,ASG0011k06,ASG000aqd9,ASG0000ryh,ASG0000s87,ASG0008q3x,ASG00…
149	hyenaSpotted	ASG0000suv,ASG0006570,ASG000n4zw,ASG000sh6l,ASG000snet,ASG0006g8h,ASG00…
150	hyenaStriped	ASG00041tb,ASG00008ta,ASG000520y,ASG00070dv,ASG0003rd4,ASG000tqqi,ASG000…
151	impala	ASG0008l0r,ASG0011qw9,ASG00032i6,ASG000n244,ASG000rs4q,ASG000e8n2,ASG00…
152	jackal	ASG0000wun,ASG0003k6o,ASG0003uq0,ASG0000wur,ASG0001mby,ASG000y3sa,ASG…
153	koriBustard	ASG000c463,ASG0006i0x,ASG0003rb7,ASG0006if5,ASG0008oik,ASG0003k76,ASG000…
154	leopard	ASG000muxj,ASG0007qno,ASG00024zv,ASG0003ldz,ASG000ssu1,ASG000dcbb,ASG00…
155	lionFemale	ASG0003uuu,ASG000r7y5,ASG0001cqw,ASG00038hl,ASG00083jc,ASG000omwr,ASG00…
156	lionMale	ASG0005mgp,ASG0006aw0,ASG0004ygj,ASG0005mfa,ASG000qlj6,ASG000ttzh,ASG00…
157	mongoose	ASG0002hc5,ASG00003df,ASG000ai54,ASG0010pxf,ASG0010pxg,ASG000p9ic,ASG001…
158	ostrich	ASG0008o7y,ASG000myqb,ASG000s89g,ASG000s89h,ASG000t83h,ASG0008l2x,ASG00…

- For each user, we identified their two top species from TD-14, then picked 20 images at random (being careful to avoid duplicates) from the per-species sets shown above [ñRECOMMEND SUBJECT SETS PER USERò in diagram above]. This combination of 40 recommended images constitutes the subject recommendations for that user [ñSUBJECT RECOMMENDATIONSò in diagram above], which can then be used in a real world scenario such as TP-506-04 below.

RECOMMEND SUBJECT SETS PER USER - Pseudocode

```
for each user
  create a subject recommendation set
  pick the two species with highest score from TD-14
  for each favourite species
    pick 20 random subjects from the subject set for that species
    add these subjects to the recommendation set for that user
  next favourite species
next user
```

Testing approach

Testing again focussed on data quality - ensuring the subject recommendations contained the userôs preferred species. This was done as follows: We randomly selected 5 users and inspected their subject recommendations. We then looked at the user profile data (species preferences, TD-14) and inspected a random sampling of at least 20 subjects for each

user. We confirmed that the subjectsô primary species were those species occurring in the userôs two favourite species. This test was successful.

The KPIs planned (precision, recall, F1-measure) were not tested since we are no longer in a comparison to external known good data scenario.

The ñchanged behaviourò borderline case was tested in TP-506-03 below, see the next section.

TP-506-03 - Test that different species preferences result in the correct changes to subject recommendations

This test focussed on ensuring that changing user species preferences would result in a correctly updated set of subject recommendations.

The approach here was to create a modified version of the TD-14 user profiles (the modified version is referred to as TD-15). This was again done using a combination of MySQL queries and Python scripts.

For example, we would pick a user whose favourite species were zebra and giraffe, and switch them to lion and gazelle.

We then repeated the tests described for TP-506-02 above, but using TD-15 instead of TD-14, and inspected the recommended subjects to ensure the ñmodifiedò favourite species were present. This test was successful.

The KPIs planned (precision, recall, F1-measure) were not tested since we are no longer in a comparison to external known good data scenario.

TP-506-04 - Real-world tests using recommender results

The main purpose of building the subject recommendations was for use in this real world experiment, referred to as the ñHappy User Experimentò. This theorized that delivering users a subject set to classify that consisted more of ñpreferredò species than random species would result in greater engagement (defined in terms of session length and number of classifications per session).

The experiment consisted of splitting all visitors to Snapshot Serengeti into two groups - the control group (which continued to receive random subjects) and the ñinterestingò or experimental group which instead received a randomly-ordered subject set consisting of:

- 20 subjects recommended to match the userôs top preferred species
- 20 subjects recommended to match the userôs second preferred species
- 30 random subjects

So in essence, each user received 40 subjects chosen by the recommender system described above, and 30 random subjects. The subject set size of 70 subjects was chosen as it was sufficiently larger than the average session length of 23 subjects.

Further details on the experimental approach are outlined on the MICO blog here: http://www.mico-project.eu/snapshot-serengeti-season-8-ready-for-mico/

The experimental findings were published in detail on the MICO blog here:
http://www.mico-project.eu/snapshot-serengeti-an-unexpected-discovery/

We found that contrary to expectations, the average session length, and average classifications per session, were much lower for the experimental/ñinterestingò cohort. This showed us conclusively that giving users more of their preferred species does not have a positive impact on engagement; or at least, not with this subject set make-up.

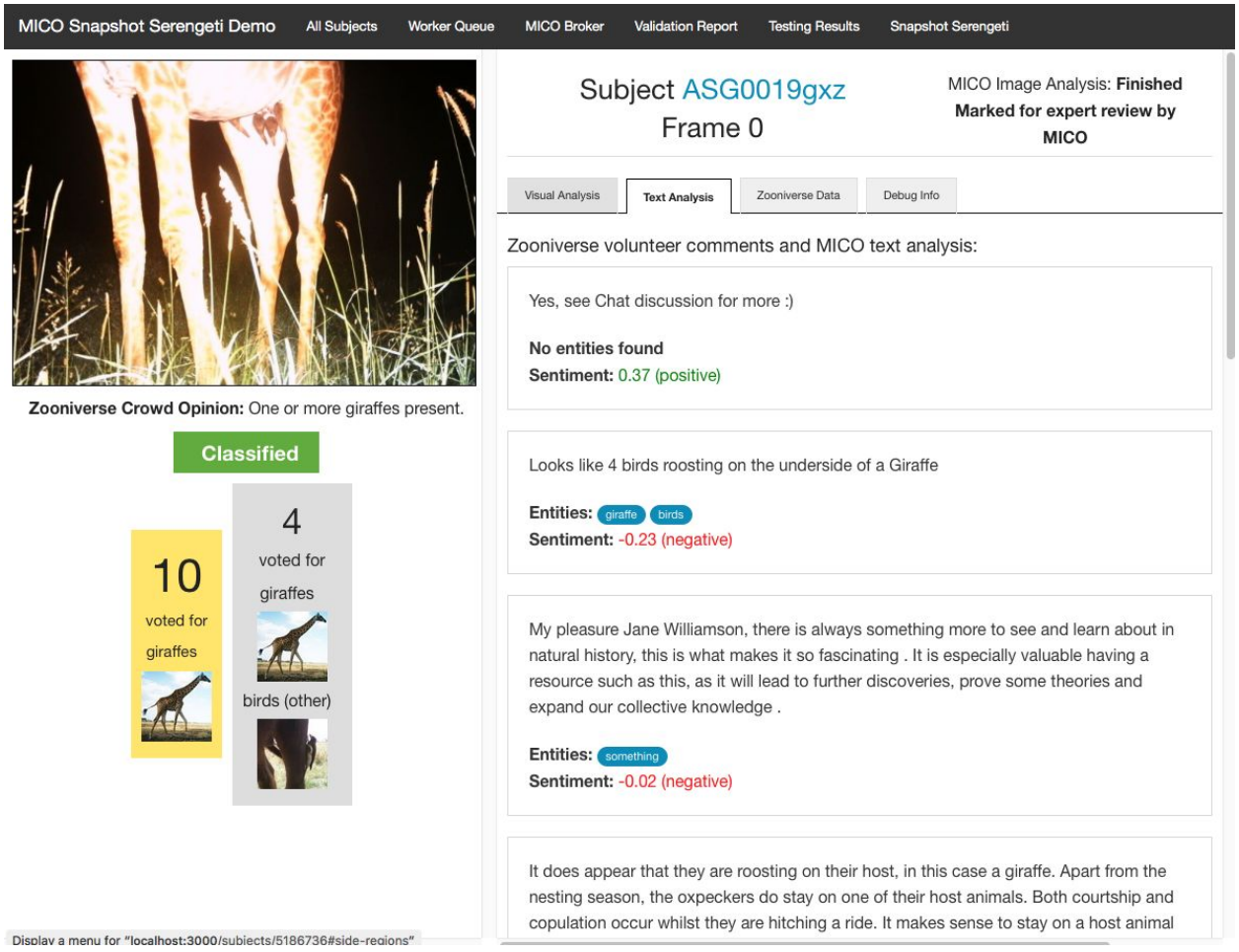
This lead us to theorize that the reason for the lower engagement was that the ñinterestingò cohort users received fewer ñblankò (no animals present) images, and that blank images play an important part in encouraging users to continue their classification efforts, probably much more than any effect of personalizing species to users. We went on to conduct further experiments into the impact of blank images; work-in-progress findings were be published at HCOMP 2015, and are available here:

http://bit.ly/blanks-paper
http://bit.ly/blanks-poster

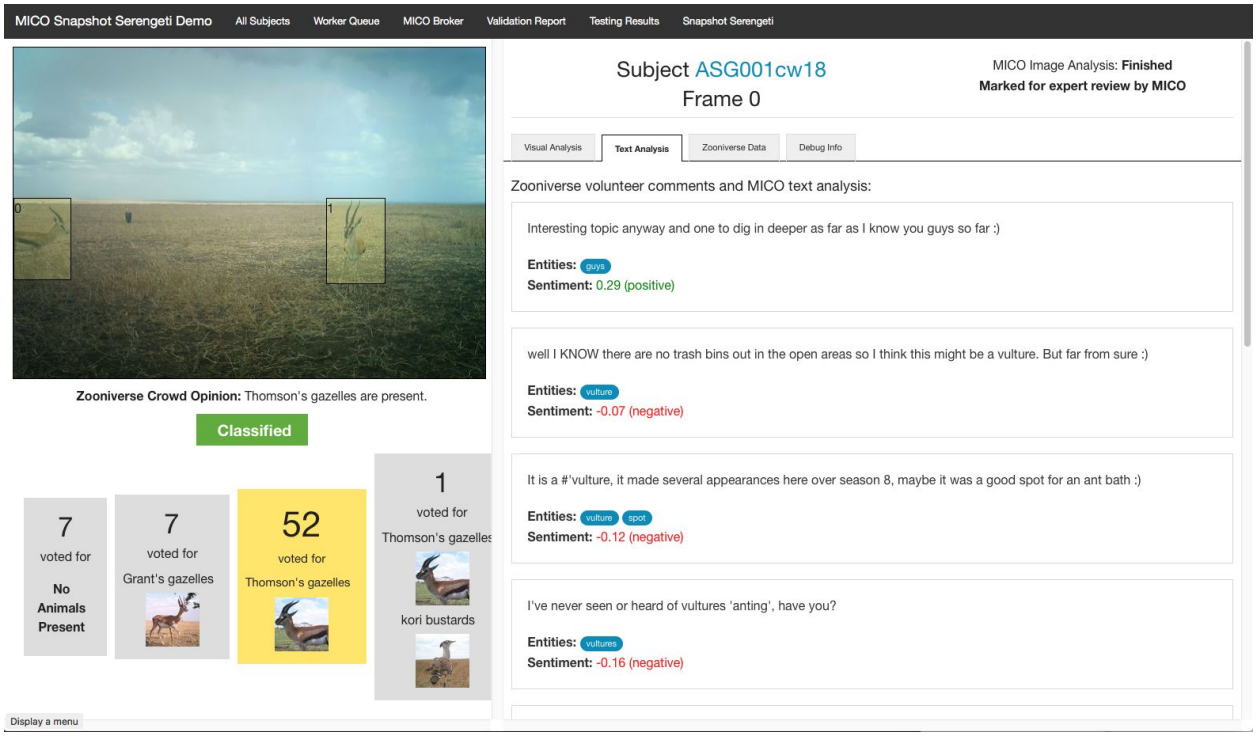
TP-508-01 - Test subjects recommended

The goal of this test plan was to assess the accuracy of the WP5 recommendation results. The recommendation engine built for WP5 takes as input extraction results for an image and the associated comments from the Zooniverse Talk forum. It then performs cross-media analysis to determine if a given image was ñdebatedò. For this, it combines detected animals from the computer vision pipeline, detected entities in the comments, and the sentiment analysis of those comments.

The recommendation engine became available towards the end of the project, and there was not enough time left to perform a full evaluation as set out in the test plan. However, we are seeing really promising results. For instance, in the image below our crowd opinion came out to only the giraffe, and while some people marked the birds nesting on the underside of the giraffe, not enough volunteers did to get it into the aggregated result.



Another example is the following image where people identified the gazelles, as did the YOLO extractor. However, both missed the animal in the distance. Some people did spot it, but because volunteers couldn't agree on what animal it specifically was, it got left out in our final aggregate answer.



Both of these things were marked as ñs debatedò by the cross-media extractor, because people were talking about them in the comments.

In our case, we could take these ñdebatedò subjects, and hand them off to a group of experts.

TP-508-02 Real-world tests using recommender results

Due to the time the WP5 recommendation engine was released, there was not enough time left in the project to run this planned evaluation.

B1: Use Cases: First Prototype - Video News Showcase - Test Plan

Goals

We aim to compare each MICO TE prior to end-to-end testing the system. For each MICO TE included in these tests we asses:

1. **outputs accuracy** - how much each single response is accurate, detailed and meaningful in terms of precision/recall and eventually where MICO TE stand when compared to other similar technologies available in the market;
2. **technical performances** - how much time each single task requires and how scalable the solution is when we increase in volume the amount of contents being analysed;
3. **usability** evaluated both in terms of **integration** and **modularity** when we add these technologies to our existing workflows.

Technical performances will be measured in terms of:

1. **latency** - time required to perform a single task on a given dataset. Measures will be repeated 10 times;
2. **scalability** - it the given TE suitably efficient and practical when applied to large situations (e.g. a large input dataset and / or, a large number concurrent requests). External platform as LoadImpact **[REF-LI]** could be used.

Usability

The TE usability requires a qualitative evaluation which will consider:

- ŏ **integration** - how simple it is to integrate each single TE within pre-existent application workflows;
- ŏ **modularity** - how simple it is to configure each TE and/or a combination of multiple TEs in a chain from within pre-existent application workflows.

Test Implementation Plan

ID	Test	Description
TP-204-01	**Test in-front face detection on images**	ŏ use dataset **TD-01** ŏ process **TD-01** with TE-204 ŏ calculate precision, recall and F1-measure on TE-204 outputs. ŏ process **TD-01** through **BetaFaceApi** [REF-BS-01] ŏ calculate precision, recall and F1-measure on TE-204 outputs. ŏ compare **KPIs**
TP-204-02	**Test lateral face detection on images**	ŏ use dataset **TD-02** ŏ process **TD-02** with TE-204 ŏ calculate precision, recall and F1-measure on TE-204 outputs. ŏ process **TD-02** trough **BetaFaceApi** [REF-BS-01] ŏ calculate precision, recall and F1-measure on TE-204 outputs.

		ǒ compare **KPIs**
TE-214-01	**Test ASR on videos containing voiceover in english without noise and/or music**	ǒ use dataset **TD-05** ǒ process ASR on **TD-05** via TE-214 ǒ calculate **Word Error Rate (WER)** on **Automatic Speech Recognition (ASR)** results transcriptions in D-05 as reference. ǒ process ASR on **TD-05** via **Vocapia** [REF-BS-03] **ǒ** calculate **WER** on ASR results transcriptions in D-05 as reference. ǒ compare **KPIs**
TP-WP05-01	**Item similarity based use case test**	ǒ use dataset **TD-03** ǒ train WP5 on **TD-03** ǒ query WP5 to generate recommendation for each item in D-03 ǒ calculate precision, recall [REF-PR] and F1-measure [REF-F1] for each WP5 outputs.
TP-WP05-02	**User based use case test**	ǒ use dataset **TD-04** ǒ train WP5 on **TD-04** - use 70% as training set, 30% as test set ǒ calculate precision, recall [REF-PR] and F1-measure [REF-F1].

B2: Use Cases: First Prototype - Video News Showcase - Validation Report

Overview of Test Results

ID	Test	Summary of Test Results
TP-204-01	**Test in-front face detection on images**	ŏ We have seen in this second evaluation **MICO outperforming BetaFaceAPI** on the precision (49% vs 43%) while having the same level of performance in terms of recall (50%) - the results have not changed since the first evaluation. ŏ We have kept the same dataset that while small helped us benchmark the results against the first validation. We can confirm that MICO (adequately trained) competes with commercial 3rd party cloud services. ŏ F-1 sees MICO at 49% and BetaFaceAPI at 46% ŏ Both algorithms struggled to differentiate the drawing of a face (from a t-shirt of one of the supporter) from real human faces ŏ The infrastructure for integrating MICO with WordPress and HelixWare has been updated and contributed: ʒ https://github.com/insideout10/mico-gateway ʒ https://github.com/insideout10/helixware-mico-plugin these modules have been designed for TP-204-01 and TP-204-02 and can also be used in TP-214-01 ŏ The experiment has produced a new tool that helps gather ground data and execute this kind of tests - this tool also has been updated for this second validation: ʒ https://github.com/insideout10/facedetection
TP-204-02	**Test lateral face detection on images**	ŏ In this case precision and recall gets lower for both as faces are harder to detect. MICO has showed precision at 13% and recall at 13%, BetaFaceAPI performed better with precision at 19% and recall at 20% ŏ The model for face detection of MICO has been trained with frontal faces (hence this result was somehow expected)
TE-214-01	**Test ASR on videos containing voiceover in english without noise and/or music**	ŏ According to the tests MICO has an average WER (Word Error Rate) of 0.925 while Vocapia (a commercial provider of speech to text cloud services) has 0.627 ŏ The WER rate does not have a great impact on the Entity Detection ŏ As with other extractors training data (which may be acquired with more contents and time), the WER value of MICO can substantially improve ŏ Since MICO supports multiple extractors, it is able to automatically extract the audio track from the ingested video assets without requiring any preprocessing from the client side (which is required by Vocapia which only accepts audio tracks of specific codecs and formats) ŏ As noted by Bjº rklund, Cleophas and Karlsson the use of probabilistic lexicalized tree insertion grammars for automatic speech recognition, in addition to the transcription produced by

		the Kaldi extractor could produce a parse structure valuable for our use case. This work is described in a paper by Bjº rklund, Cleophas and Karlsson, presently to be submitted for publication.
TP-WP05-01	**Item similarity based use case test**	ð WP5 components produce plausible user-customized recommendations ð Proposed test metrics for recommendation use cases should be improved.
TP-WP05-02	**User based use case test**	ð WP5 components produce plausible user-customized recommendations ð Proposed test metrics for recommendation use cases should be improved.

Details Test Reports

We have revised the components used for the integration between MICO and our software tools HelixWare and WordLift in order to adapt to the new MICO broker.

The tests during the evaluation phase did not provide any change in terms of quality of the extractors as the major update in the MICO platform for year 3 involved the broker and not the cross media analysis tools. The most significant update is related to the possibility provided by the new broker to choose from the MICO API the pipeline to be used for the analysis.

At the technical level the changes we have done to the integration components are the following:

- ð inject/add request:
  - ȝ (1) the parameter ócióhas changed into ótemUriô,
  - ȝ (2) the parameter ótypeôused to be a mime type (i.e. image/png) and now is a MICO type (mico:Image),
  - ȝ (3) the mime type is not longer required
- ð inject/add response:
  - ȝ The format of the response has changed from óuriôto ótemUri'
- ð inject/submit:
  - ȝ (1) the parameter ócióhas changed into ótemô
  - ȝ (2) the parameter órouteôhas been added - this provides a way to programmatically chose the analysis chain (i.e. route=3 identifies the route for the face detection)

The updates in the data model also reflected the required updates for the query used to get the metadata out of the system. In the case of the face detection for instance the query has been updated as shown below:

Previous Query for face detection (MICO Y2)

```
PREFIX mico: <http://www.mico-project.eu/ns/platform/1.0/schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dct: <http://purl.org/dc/terms/>

SELECT DISTINCT ?region WHERE {
<${header.MicoItemUri}> mico:hasContentPart ?cp .
  ?cp mico:hasContent ?annot .
  ?annot oa:hasBody ?body .
  ?annot oa:hasTarget ?tgt .
  ?tgt  oa:hasSelector ?fs .
  ?fs rdf:value ?region
  FILTER EXISTS {?body rdf:type mico:FaceDetectionBody}
}
```

New Query for face detection (MICO Y3)

```
PREFIX mmm: <http://www.mico-project.eu/ns/mmm/2.0/schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX oa: <http://www.w3.org/ns/oa#>

SELECT DISTINCT ?o WHERE {
 <${header.MicoItemUri}> mmm:hasPart [
    mmm:hasBody [ a
<http://www.mico-project.eu/ns/mmmterms/2.0/schema#FaceDetectionBody> ] ;
    mmm:hasTarget [ oa:hasSelector [ rdf:value ?o ] ]
] }
```

The data model has been streamlined and we have done some improvements on the queries to increase their readability and reuse within our team.

Throughout the testing we had seen the following:

1. downtime and stability issues have been greatly reduced compared to the previous year while the information provided to troubleshoot the issues is still to be improved the overall feedback is positive.
2. As for last year validation tests and integration work has been delayed as the early versions of the updated platform had issues that needed to be fixed before our testing could actually being. The extracted metadata had a slightly different format compared to last year and we had the opportunity to revise our querying approach.

Background for TP-204-01 and TP-204-02: Face detection

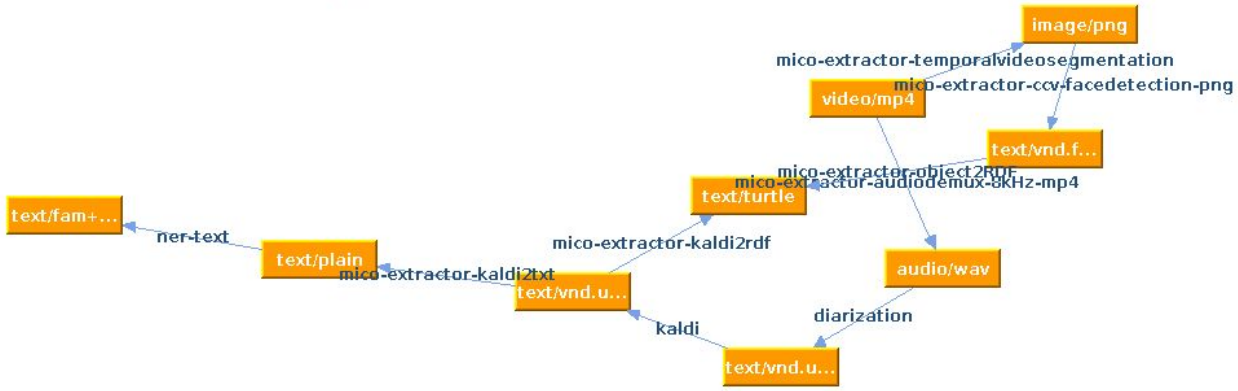
We began evaluating MICO in the context of media curation and filtering for two of our main use cases: the Greenpeace magazine and Shoof the micro-recording video app.

The basic idea behind this test is that for organisations being a non-governmental organization (NGO) like Greenpeace or a startups like Insideout Today (the company developing Shoof) a great effort is needed to curate and filter media assets.

Face detection is an important part of face recognition as the first step of automatic face recognition. Face detection allows us to classify content that contains human beings and also to build interlinking between audiovisual and textual contents that share the same people (and possibly the same topic).

However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating condition and facial expression. In this test we've been focusing our attention to front face detection.

While our goal is to detect faces in videos (and we have two pipelines in MICO supporting this requirement) the test is based on a small dataset of images derived from the Digital Asset Manager of Greenpeace.



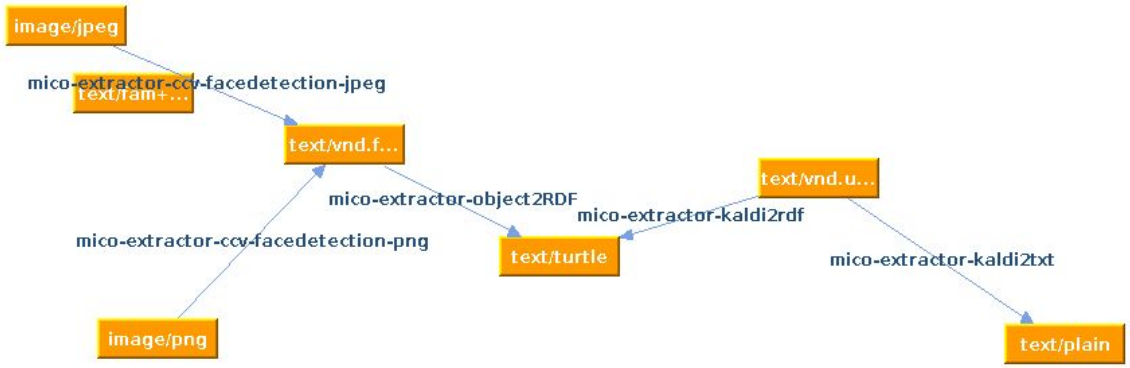
Pipeline combining temporal video segmentation, face detection and (ASR) automatic speech-recognition - takes in input a video

The analysis chain that combines face detection, temporal video segmentation with ASR works as follows:

1. audio-videos files are ingested (in our case from a WordPress front-end or via APIs)
2. the audio part is sent to Kaldi for the Automatic Speech Recognition
3. while the video part is sent to the temporal video segmentation
4. and the resulting images are processed by the face detection extractor

This pipeline is extremely complex, as explained above, we've decided instead to use images for the evaluation rather than videos. Using images helped us compare MICO with other systems and provided a quantifiable measurement for the testing.

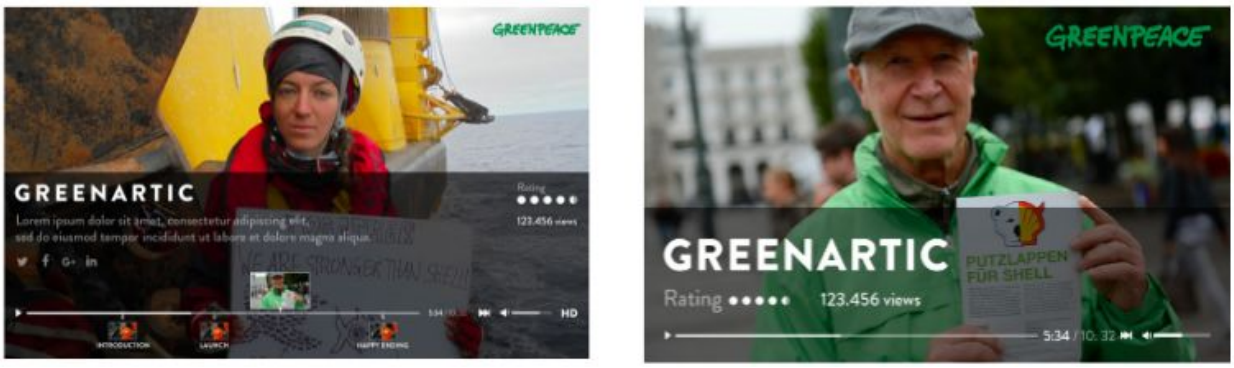
Here follows a diagram representing the pipeline used for the testing (as seen from the MICO broker admin interface).



Pipeline providing face detection on images

The face detection extractor was deployed starting from June 2015 and had subsequent releases. The extractor, developed by the Fraunhofer team, is using an open source library (CCV - http://libccv.org) and has been trained for in-front faces so we expected better results on TP-204-01 rather than on TP-204-01 (that first test is for in-front faces and the second is for lateral faces).

During this year we have also begin designing a new use case to leverage on face detection to improve the user engagement on the Greenpeace magazine.



The new HelixWare video player powered by MICO

The annotations provided by MICO are in the form of media fragments and can be used as input for both the embeddable video player of HelixWare and the video thumbnails created by HelixWare WordPress plugin. Media fragments, in this specific case, are generated from the

face detection extractor and are both temporal (there is a face at this time in the video) and spatial (the face within these frames is located at xywh). The new HelixWare video player that we are developing aims at creating an immersive experience for the end-users. The validation of both video player and video thumbnail will be done using A/B testing against a set of metrics our publishers are focused on: time per session, numbers of videos played per session, number of shares of the videos over social media (content virality).

Results Summary

In the tests the definitions used are:

- ǒ True positive: faces correctly identified (including any intersects)
- ǒ False negative: faces incorrectly identified as not present
- ǒ False positive: faces incorrectly identified (intersect = 0)
- ǒ Precision: the percentage of faces identified by the system that are indeed faces (intersect > 0%)
- ǒ Recall: the percentage of correctly identified faces among the total number of faces that are present in the dataset.

Here is a snapshot of the results of the test accuracy in terms of precision, recall and F-1 measure **combining in-front and side face detection - this has not changed since last year**:

BetaFaceAPI

Precision	Recall	F-1
0.2903225806	0.3214285714	0.3050847458

MICO

Precision	Recall	F-1
0.2743362832	0.2767857143	0.2755555556

MICO for in-front face detection outperforms BetaFaceAPI and proves to be immediately usable in our showcase. The side detection of MICO is poorer than BetaFaceAPI (therefore the combined results are affected) - in both cases, when dealing with side faces, the complexity is much higher and the overall accuracy is low.

Evaluation Approach

In order to asses the quality of the analysis provided by MICO we need to know if and where each face is positioned within each image. Moreover, as a startup aiming to bring these functionalities in our product line, we want to make sure the quality of the results is comparable to existing solutions available in the commercial world.

As we needed to manually annotate the dataset composed of 60 images (30 for in-front faces and 30 for later faces) and to confront the result of MICO with the results from a commercial provider we decided to develop a web tool for automating the task.

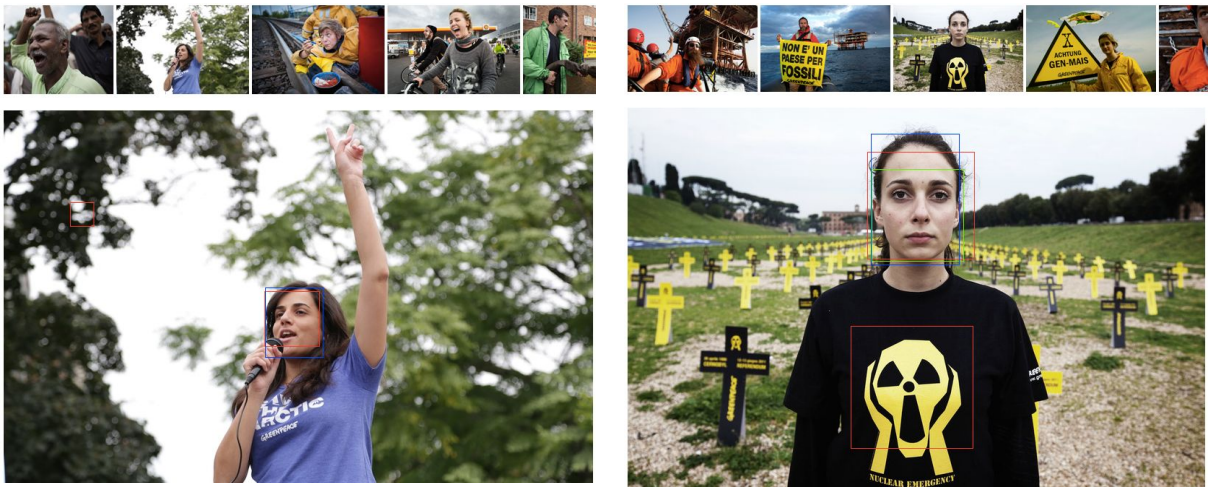
While for a statistical robust evaluation we would have needed a much larger dataset the changes in our evaluation strategy (from testing in an integrated / live scenario to a classical evaluation approach) had forced us to keep the number of items very low at this stage. Building a web tool, on the other end, was a good move as we can now scale up these datasets and continuing the evaluation work on the next platform iterations.

The tool has been contributed as open source and is available on GitHub - the tool as been updated to function with the latest release of MICO:

- https://github.com/insideout10/facedetection

The third party service providers we've used for the benchmarking is BetaFaceApi (http://betafaceapi.com) a commercial SaaS offering face detection and face recognition in the cloud (their pricing starts at `0.035 EUR` per image).

Here are two screenshots of the tool we created for the testing.



The tool has been designed with the purpose of executing the following tasks:

1. upload the dataset of images
2. manually annotate the images using a rectangular-shaped selector - the color of the manual annotation is blue
3. sending each image to BetaFaceAPI and to MICO
4. handle exceptions from MICO and retry sending the image after 3 seconds
5. receive the analysis results from BetaFaceAPI and add a red annotation on each image
6. receive the analysis results from MICO and add a green annotation on each image

7. store all corresponding media fragments as separate image files
8. store the coordinates of each annotation (manual, BetaFaceAPI and MICO)
9. calculate overlaps, KPI test results and export it to CSV

The tool uses:

- backbone.js for the manual annotations
- Apache Camel for handling the analysis over to BetaFaceAPI and MICO
- HATEOAS (Hypermedia as the Engine of Application State) to navigate the REST APIs

For further refining the results of this test we are also planning to update the tool for implementing a new existing definition of True Positive in order to have:

- intersection (region_annot, region_gt) / union (region_annot, region_gt) >= 0,5

The results of this evaluation are also accessible on: http://facedetection-mico.insideout.io.

Test Framework Used

We used the MICO Platform API that was setup and managed by the team at Salzburg Research. See http://mico-project.bitbucket.org/api/ or document D6.2.2 for further details of custom animal detection and text analysis endpoints.

Using this API, the IO10 team has worked in three directions:

1. implement MICO functionalities in our existing workflows for HelixWare and WordLift
    a. this has provided an in-depth understanding of the existing strength and limitations of the platform in terms of integration effort required by a third party willing to leverage on MICO existing functionalities
2. engaging with our existing clients to understand at best their requirements and how MICO can bring concrete advantage within their operations - Greenpeace Italy on one side and the Shoof team on the other
    a. on this front, the complexity of the analysis pipelines we use, the overall maturity of some of the querying tools and the overall instability of the system, as presented above, limited the degree of integration between MICO and our software modules
3. test and evaluate the accuracy of the extractors being used in our showcase accordingly to the test plan designed within WP7 and WP8

TP-204-01 - Test in-front face detection of images

For this test we wanted to asses the quality of the annotations provided by MICO for Face Detection against a dataset of images from the Greenpeace DAM containing in-front faces.

The annotations derived from MICO and BetaFaceAPI have been analyzed against manually annotated faces (with a margin of error +/- 0.01%).

Metrics for the in-front face detection set

For testing the correct detection of a face, we defined the metrics below:

- Relevance is defined as ñThere is a faceò
- Intersection is defined as ñThe system annotation intersects with the manual annotationò
- This leads to the following definitions:
  - True positive: faces correctly identified (including any intersects)
  - False negative: faces incorrectly identified as not present
  - False positive: faces incorrectly identified (intersect = 0)
  - Precision: the percentage of faces identified by the system that are indeed faces (intersect > 0%)
  - Recall: the percentage of correctly identified faces among the total number of faces that are present in the dataset.
  - F1 score: overall measure of this testôs accuracy, combining precision and recall

- Results are available online in a Google Spreadsheet: https://goo.gl/LcALKQ
- A selection of faces detected by BetaFaceAPI is provided over here as an example: https://goo.gl/yxtKic

As we expected for in-front faces **MICO outperformed BetaFaceAPI and proved to be immediately useful in our showcase.**

Results from MICO

	Quantity	True Positive	False Positive	False Negative	Precision	Recall	F-1
Frontal	44	22	1	22	0.4888888889	0.5	0.4943820225

Results from BetaFaceAPI

	Quantity	True Positive	False Positive	False Negative	Precision	Recall	F-1
Frontal	44	22	7	22	0.431372549	0.5	0.4631578947

TP-204-02 - Test lateral face detection of images

For this test we wanted to asses the quality of the annotations provided by MICO for Face Detection against a dataset of images from the Greenpeace DAM containing side faces (usually much harder to detect than in-front faces).

As for the previous tests annotations derived from MICO and BetaFaceAPI have been analyzed against manually annotated lateral faces (with a margin of error +/- 0.01%).

Metrics for the side face detection set

For testing the same metric of TP-204-01 have been used.

- ð Results are available online in a Google Spreadsheet: https://goo.gl/LcALKQ

As the detection model behind MICO was not trained for side faces BetaFaceAPI outperforms MICO but; in both cases the accuracy (F-1) is quite low**.**

Results from MICO

	Quantity	True Positive	False Positive	False Negative	Precision	Recall	F-1
Side	68	9	0	59	0.1323529412	0.1323529412	0.1323529412

Results from BetaFaceAPI

	Quantity	True Positive	False Positive	False Negative	Precision	Recall	F-1
Side	68	14	5	54	0.1917808219	0.2058823529	0.1985815603

Background for TP-214-01: Automatic Speech Recognition

The main goal of TE-214 Automatic Speech Recognition (ASR) in our showcase is to be able to interlink audiovisual contents with text articles.

An editor using WordLift (a semantic editor for WordPress that is being used by Greenpeace for their online magazine website) has access to a large amount of video contents via the internal digital asset manager. The amount of time required to properly filter and select a video to enrich an article is though too high and in most cases incompatible with the tight publishing schedule of the organisation.

Also for the editorial team that will work behind Shoof (the micro video-recording application) filtering and moderating user-generated contents can be extremely costly and time consuming. The use of ASR in MICO for our showcase is therefore not intended to generate readable transcripts but to help editors working with Content Management System and Digital Asset Managers quickly scan, filter and interlink media files with other contents (primarily news).

Evaluation Approach

As for the face detection also for ASR it is important to us to ensure the quality of the results of MICO is somehow comparable to what we can achieve with commercial service providers offering this functionality in the cloud.

Following the expert advice of MICO transfer partner CodeMill AB for this test we ha chosen to compare MICO analysis results with Vocapia (http://www.vocapia.com/) a provider of speech-to-text software services.

The testing has been done using an open source Java class that aligns two string sequences and calculates the word error rate (WER) - the metrics we've chosen for this test. The library is available on https://github.com/romanows/WordSequenceAligner.

Metrics for testing ASR

The tests used the Word error rate (WER), a common metric of the performance of a speech recognition or machine translation system. The general difficulty of measuring performance lies in the fact that the recognized word sequence can have a different length from the reference word sequence (supposedly the correct one).

The WER (as by definition of Wikipedia) is derived from the Levenshtein distance, working at the word level instead of the phoneme level. The WER is a valuable tool for comparing different systems (MICO and Vocapia in our case) as well as for evaluating potential improvements within TE-214 (the Kaldi extractor responsible for the automatic speech recognition in MICO).

TP-214-01 - Test ASR on videos containing voiceover in English without noise and/or music

Here follows the list of files in **TD-05**. We've chosen a selection of videos with subtitles in English stored in a separate .srt file. These videos have been taken from the Greenpeace

worldwide digital asset manager. While each video contains some voiceover in English it was hard to find videos completely without noise or music.
The size of the dataset is relatively small in this case as for TP-204-01 and TP-204-02.

Test results

It is worth mentioning that while in MICO the extractor is using a standard model for English the ASR provided by Vocapia had a specific settings for British English.

ID	A/V Filename	CC Filename	MICO WER	Vocapia WER
1	An_upload_from_Santa_himself_Christmas_2013_might_be_cancelled	An_upload_from_Santa_himself_Christmas_2013_might_be_cancelled_audio.ref	1.7361112	1.0833334
2	BREAKING_Shells_Arctic_oil_rig_in_sight	BREAKING_Shells_Arctic_oil_rig_in_sight_audio.ref	0.6785714	0.4945055
3	EXPOSED_Santander_loaned_millions_to_the_company_that_did_this	EXPOSED_Santander_loaned_millions_to_the_company_that_did_this_audio.ref	0.810219	0.5985401
4	Fukushima_Four_Years_On	Fukushima_Four_Years_On_audio.ref	0.8531856	0.66759
5	Ice_Ride_the_global_bike_ride_to_defend_the_Arctic_Are_you_in	Ice_Ride_the_global_bike_ride_to_defend_the_Arctic_Are_you_in_audio.ref	0.9219858	0.6312057
6	Living_For_A_Cause_1_Kumi_Naidoo	Living_For_A_Cause_1_Kumi_Naidoo_audio.ref	0.8073879	0.66490763
7	Stephen_Tindale_Former_Greenpeace_UK_Executive_Director_Nuclear4Climate	Stephen_Tindale_Former_Greenpeace_UK_Executive_Director_Nuclear4Climate_audio.ref	0.8577778	0.5688889
8	Thank_you_from_Greenpeace_volunteers_on_Shells_rig	Thank_you_from_Greenpeace_volunteers_on_Shells_rig_audio.ref	0.88142294	0.66007906
9	The_Rainbow_Warrior_comes_to_London	The_Rainbow_Warrior_comes_to_London_audio.ref	0.90618765	0.7205589
10	The_Root_of_the_Matter_FSC_Greenpeace_and_Intact_Forest_landscapes	The_Root_of_the_Matter_FSC_Greenpeace_and_Intact_Forest_landscapes_audio.ref	0.79647434	0.6383547
		Average	0.924932363	0.672796389

The speech-to-text analysis results are available here:

- https://goo.gl/ywCypa

Latency

In our environment videos are sent to HelixWare (the online video platform) and from there to MICO for content analysis. As different analysis pipelines can be chained one to the other (hence increasing the amount of time required for analysing an asset) we have implemented a job queue via AMPQ. This way HelixWare calls MICO with batch processes that run independently and asynchronously.

Whenever an asset is received, or by user request, HelixWare starts a job configured to run a specific task and the task might be simple or complex (e.g. a Speech-To-Text task or a sequence detection task).

The job tracks the status, whether the activity has started, is progressing, is completed and results have been received; it also deals with multiple nodes, which one is going to handle the job, whether distributed communication is involved.

The task represents the actual data protocol which is required in order to provide something to process and what kind of response is expected (e.g. for a speech-to-text task, a response is expected that contains the transcribed text along with markers about the timecodes to sync the text with the video and eventually who's speaking).

Due to the blocking problem related to the missing of status information from MICO (no means of understanding when an analysis complets) we were not able to asses - during our evaluation tests - latency measured as the time interval occurring between the submission of the image and the reception of the analysis results.

Throughput

We have seen MICO coping with multiple parallel submissions, which means that in practice we could send up to 3 videos in a row without crashing the system. From the architecture point of view the MICO broker seems capable of scaling.

Stability

During the testing for this evaluation, MICO stability greatly improved from last year.

Modularity

With the new broker we can now use MICO cross-media extractors in a more flexible way as various routes can be configured at the application level.

Integration

We decided to evaluate the integration of MICO in our existing toolset and we've been documenting the code required for interfacing MICO in terms of:

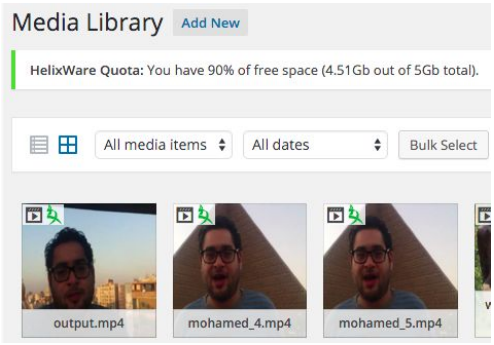
- Publish content items to MICO
- Start processing the content item
- Get an initial status report on the processing state
- Extract an initial set of metadata to be integrated within the application

These steps are fully documented here:

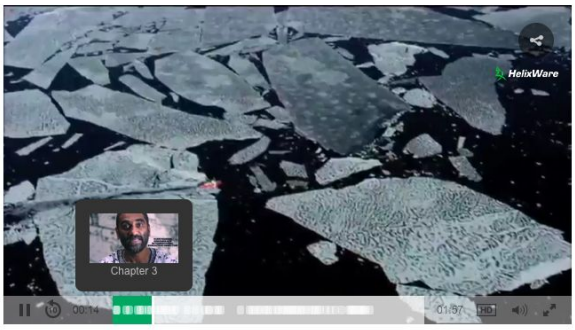
- https://goo.gl/UmjecN

As our original plan was to evaluate MICO in the real-world showcases we've built the following modules to integrate MICO with HelixWare and WordPress. The following contributions are open source:

- **MICO-Gateway** (https://github.com/insideout10/mico-gateway): based on the steps described above and for the purpose of interfacing MICO with HelixWare we've developed this component that can be used for helping others integrating with MICO in Java.
- **HelixWare MICO extension for WordPress** (https://github.com/insideout10/helixware-mico-plugin): this plugin extends the existing integration of HelixWare with WordPress to bring the result of the analysis privided by MICO to the CMS editor. The plugin is designed for the Greenpeace magazine use case and might also use for Shoof.



1. Media assets created with Shoof appear in the standard WordPress Media Library and from there are sent to HelixWare for encoding and to MICO for analysis.



2. The player embedded in the webpage using a WordPress shortcode displays the video including the chapters resulting from MICO TE-206 (temporal video segmentation). Player has been updated since last year

Usefulness

We expect to leverage on the the results on MICO to extend the functionalities of our online services and software. We strongly believe RTD effort like MICO shall bring concrete advantages to startups like ours and to the developer community at large.

We will therefore continue to do our best to make this happen. As of today we have seen some concrete values in the provided extractors and the stability of the platform is greatly improved.

The overall platform and the testing we have done represents an overall positive outcome for MICO.

TP-WP05 - Item and User based use case tests

One part of the MICO recommendation work package is the recommendation based on item and user statistics: TP-WP5-01 and TP-WP5-02. During project lifetime, it became apparent that the focus will be set on providing recommendation services for the Greenpeace Magazine and services for the Zooniverse platform. Specifically, the two tasks were

- ð Providing article recommendation to users of the Greenpeace magazine
- ð Providing recommendations for favourite images within Snapshot Serengeti (Zooniverse)

Both tasks have been implemented within MICO, see *D5.3.2 Enabling Technology Modules: Final Version* for details.

The test plan distngushes between recommendations based on items and recommendations based on users / usage information. However, both can be covered by collaborative filtering only: Recommendation for registered users (mapping user-ids to item-ids) and recommendation for unregistered users (mapping item-ids to item-ids). The user-based recommendation is used as a base function, the item based recommendation is interpreted as the succession of searching a user that has positive interest in the current item, and recommend items based on his profile.

The test plan provides for a measurement of precision and recall (and F-measure as a combination of both) on a split test data set. However, recommendation problems have a slightly different nature than classification problems, which makes it hard to apply those metrics. Both precision and recall are hard to determine. The process of recommendation is to find *something* that is *interesting* to the subject. This may change over time, even during the time the user is reading an article. As measuring the recall requires to know the amount of items that were interesting, but have not been recommended is even for small item sets a challenging task.

Similar arguments are valid for the evaluation of the precision of a recommender. Usually, the recommendation logic is re-used for validation, the recommendation system clusters item using a certain set of rules. At validation stage, it is tested that those rules still apply. Due to the subjectivity of the recommendations, those rules will never be 100% correct. In other words: it is very hard to evaluate recommendation systems offline, as it is not possible to get reliable ground truth data for it. Even assuming that there is such data, the popular scheme of k-folds validation is problematic, as the input data consist of time series data with data points that are highly dependent. Still, the evaluation of the recommender system is possible, but not using the proposed 70%-30% cross validation, but by measuring the recommender quality indirectly using user studies as discussed in section *8.2.3 Online Evaluation* of the *Recommender Systems Handbook* by Ricci et.al.

While it was not possible to run such field studies during the project time span, they have been prepared and can always be conducted when the showcase partners use the MICO platform within their products. Proof-of-concept evaluations were done in the implementation of the WP5 demos, as seen in the following screenshot, showing several article recommendations for a given user ID:

The following sections evaluate the recommendations provided by collaborative filtering using the prediction.io platform. For both Greenpeace Magazine and the Zooniverse showcases, firstly the features of the available datasets are analyzed, and secondly the results of the recommendation, within the limitations of the provided training data, are discussed.

Greenpeace Magazine - Dataset Description

This section discusses the collaborative filtering recommendation approach, implemented using TD04 (US-61). The key properties of the test data will be outlined. The following section discusses the achieved recommendation quality.
The Greenpeace Magazine dataset is a csv file consisting of 6941 lines with the following structure:

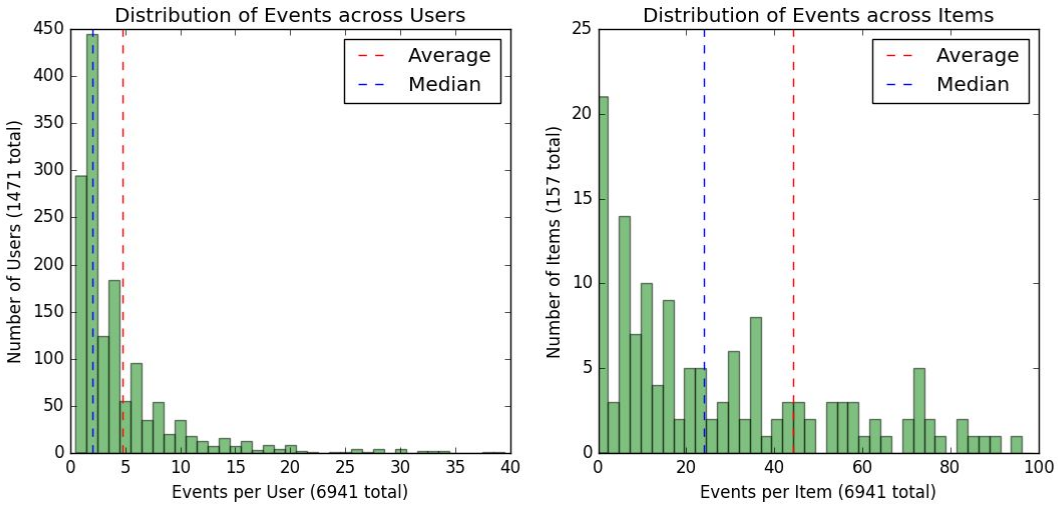
```
/micro-plastiche-grandi-guai/,view,484882
/nazareth-sanzini/,view,484882
```

```
/la-protesta-per-salvare-le-api/,view,322044
/micro-plastiche-grandi-guai/,view,322044
```

Key statistics of TD04 (Greenpeace Magazine Site Statistics)	
Number of Events	6941
Number of distinct Items	157
Number of distinct User IDs	1471
Average Events per Item	44.2
Average Events per User	4.7
Median Events per Item	24
Median Events per User	2
Maximum Events per Item	784
Maximum Events per User	81

We call each of this line an ***event***. The first value (*item*) is a subresource within the greenpeace website, most of the time articles:
http://magazine.greenpeace.it/nuove-norme-contro-il-legno-illegale/

The second value, is either *view* or *repeated view*, the third value is the *user* id. It is important to note, that these IDs are most likely session-local. So there is no tracking of registered users. That means that different user ids do not necessarily represent different users. There are no rating of articles, statistics on visit durations or similar.



Therefore, the key for generating good recommendations are overlaps between users or items, to exploit transitivity for collaborative filtering. To have ñoverlappingò profiles, there must be users that visit several items or items that have been visited by several users. Sadly, the majority of users only has one or two interactions with the site, which makes it hard for a collaborative filtering system like prediction.io to deduce meaningful results. The exact distribution of events per user and events per items are shown in the histograms. The vertical lines denote the median and the average value across the whole dataset.

Greenpeace Magazine - Recommendation Evaluation

Despite anticipating no exceptional results, we trained the prediction.io recommender with the provided data. The dataset was split in 75% training data and 25% test data, after grouping it by users. In other words: the user ids in training set and test set are disjoint.
For each users in the training set, it is assumed that the visited pages can act as ground truth data. For various reasons outlined at the beginning of this chapter, this is interpretation is problematic. However, given the data at hand is was the only choice.

For each user in the test set, one item was used as an input for the recommendation engine. If the recommended items were in the set of other visited items, this was count as a hit. With some limitations, this can be interpreted as a weak form of *precision* metric.

For different dataset partitionings tested, the hit rate was always at around **13%**. Still, as discussed above, due to the subjective nature of recommendations it is hard to interpret this value without conducting further user studies. For an overview, the following table lists a random sample of recommendations generated. The numbers are a score returned by prediction.io which was not interpreted during the evaluation but can be used by application developer as an indicator for the ranking of items.

When the MICO recommendation engine is deployed to the Greenpeace Magazine website, it is possible to measure the click rates on the provided recommendation. Using A/B testing, it will be possible to measure, how end user acceptance, i.e., recommendation quality, differs using different settings of the recommendation engine. Additionally, the recommender performance can be compared to the ñrandom recommenderò. Whether criteria like ñFreshnessòof the articles have an influence can be tested as well.
Additionally, asking a closed user group, the recommendation will help to further fine tune the recommendation. We propose to conduct this research for the final use case validation, as it was not possible to conduct those measurements now.

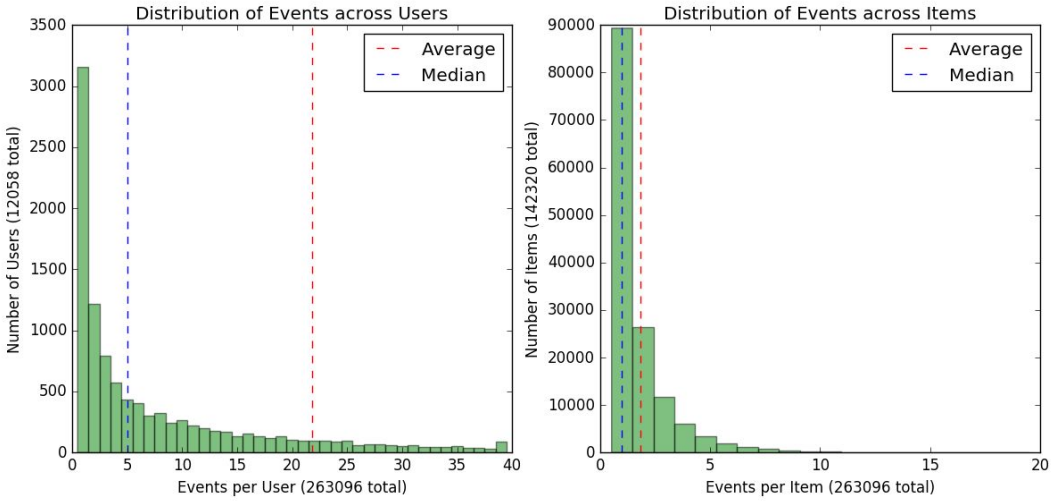
Even if the test methodology is adequate, evaluation results still might be ambiguous: The scope of articles inside the Greenpeace Magazine is very narrow. It is unlikely that there will be ñwrongòrecommendations at all, except for cases that might be ruled out programmatically (presenting articles, the user already read, presenting technically faulty items).

Recommended for /archivio-numeri/ /svolta-epocale-o-buco-nellacqua/: 1.3587315082550049 /ttip-leaks-la-trasparenza-prima-di-tutto/: 1.2616087198257446 /lagricoltura-sostenibile-in-mostra/: 1.1544746160507202 /gaia-calabresi/: 0.969948947429657	**Recommended for /quel-trattato-e-una-truffa/** /tossico-come-un-pesticida/: 3.3909857273101807 /un-altro-clima-e-possibile/: 1.6944133043289185 /gaia-calabresi/: 1.0258492231369019 /ttip-leaks-la-trasparenza-prima-di-tutto/: 0.4033706784248352
Recommended for /la-rainbow-warrior-in-africa/ /tossico-come-un-pesticida/: 1.8589996099472046 /gaia-calabresi/: 1.1767537593841553 /lagricoltura-sostenibile-in-mostra/: 0.828025758266449 /un-altro-clima-e-possibile/: 0.3659321367740631	**Recommended for /il-pesce-azzurro-nel-canale-di-sicilia/** /tossico-come-un-pesticida/: 3.60068416595459 /gaia-calabresi/: 1.1986593008041382 /un-altro-clima-e-possibile/: 1.1802020072937012 /lagricoltura-sostenibile-in-mostra/: 0.45847123861312866
Recommended for /aumenta-la-violenza-sugli-attivisti-ambientali/ /ttip-leaks-la-trasparenza-prima-di-tutto/: 1.474746823310852 /archivio-numeri/: 1.35952627658844 /svolta-epocale-o-buco-nellacqua/: 0.4381033480167389 /lagricoltura-sostenibile-in-mostra/: 0.06784950941801071	**Recommended for /il-pesce-azzurro-nel-canale-di-sicilia/** /tossico-come-un-pesticida/: 3.60068416595459 /gaia-calabresi/: 1.1986593008041382 /un-altro-clima-e-possibile/: 1.1802020072937012 /lagricoltura-sostenibile-in-mostra/: 0.45847123861312866
Recommended for /accendiamo-il-sole-il-futuro-e-adesso/ /svolta-epocale-o-buco-nellacqua/: 1.7566988468170166 /ttip-leaks-la-trasparenza-prima-di-tutto/: 1.6707779169082642 /archivio-numeri/: 1.560915470123291 /lagricoltura-sostenibile-in-mostra/: 0.41537654399871826	**Recommended for /lanno-piu-caldo/** **/svolta-epocale-o-buco-nellacqua/:** 1.495859980583191 /lagricoltura-sostenibile-in-mostra/: 1.3025898933410645 /archivio-numeri/: 1.1648626327514648 /gaia-calabresi/: 1.061357021331787

Zooniverse User Likes - Dataset Description and Recommendation Evaluation

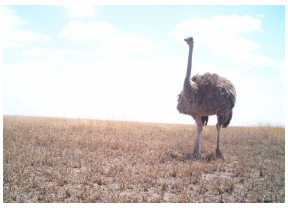
Similar to section on the greenpeace dataset, this discussion discusses the dataset for the Zooniverse *Favourite Images* recommendation (US-66b-B)

Key statistics of Zooniverse image Likes DB	
Number of Events	263096
Number of distinct Items	142320
Number of distinct User IDs	12058
Average Events per Item	1.8
Average Events per User	21.8
Median Events per Item	1 *(sic!)*
Median Events per User	5
Maximum Events per Item	92
Maximum Events per User	1000

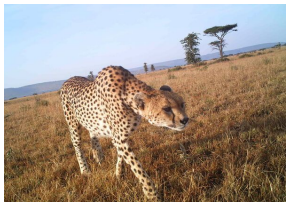


Apparently, the majority of users only have one favourite image, and only very few images are annotated by several users. The main reason for this might be in the way the Zooniverse Platform works: Only a very limited amount of users have access to a certain image. Missing a mechanism for automatically promoting ñinterestingòimagery, not a lot of users getting their attention drawn to such images.

As a conclusion, other factors might be a better source for recommendation, e.g., the number of discussion items related with a certain subject. Within MICO, such information is used in the cross-media case for debated images but not in the simple image recommendation case. However, Prediction.io was trained with the data and is able to give recommendations based on a popularity score, returning a selection with an expected appeal.

			
ASG0014cw6	ASG00182fb	ASG000a0dt	ASG00013g8

For reference, the 8 images with the most likes within the dataset are:

ASG0004jcv (92 likes)	ASG0003sz9 (63 likes)	ASG0000acf (57 likes)	ASG00181pp (51 likes)
			
ASG00181p7 (51 likes)	ASG0003ib1 (47 likes)	ASG001cmtr, (46 likes)	ASG0005npp (42 likes)

However, to implement a favourite image recommender within Zooniverse, compared to collaborative filtering with prediction io, it seems to be a better approach to take items with a lot of discussions, positive sentiment which are not debated from the classification point of view.

WP2 orchestration - qualitative validation aspects

While previous sections have provided validation information from the showcase perspective based on pre-defined workflows, many WP2 orchestration aspects cannot be reasonably validated, due to the fact that orchestration is mainly about workflow creation and execution and back-end aspects which are not visible on the showcase side.

Hence, the following includes a short discussion on our own qualitative and subjective assessment of achievements regarding extractor orchestration in WP2. It is based on the requirements that were analyzed after it became clear that broker v1 would not be sufficient for the project, aka the ñbroker wishlistò described in sec 2.7 of Dx2.2 (Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations: Final Version). We will list the individual requirements, and provide comments on whether and how they have been addressed with the final broker version:

- ð FR-2 Support for multiple output items: This has been addressed by broker v3 - each extractor can produce several types of output without disturbing routing to following extractor Ÿ OK
- ð FR-3 Support for multiple input items: Indeed, the broker can now collect new parts until all items necessary for next processing step are available before sending them to further extractors. The Event API was extended to point the extractor to multiple parts in one analysis request (list of resources, which can contain parts and item) Ÿ OK
- ð FR-4 Support for ñNo outputò Broker v1, being based on triggering extractors based on MIME-Types, always required extractors to provide new parts (in case, empty ones) to trigger other extractors. With the new broker, an extractor can extend metadata of an existing part without creating a new part to trigger the next extractor (which was what was needed) Ÿ OK
- ð FR-5 Support for dealing with (more complex) extractor dependencies: By describing an extractor via *registration.xml*, or extending such information via the *registration service*, much more complex dependencies can now be expressed and used. Input can now be binary data, a part created by another extractor or even an annotation in the metadata store, e.g. a detected language or a framerate annotation Ÿ OK
- ð FR-6 Support for status tracking: With previous broker versions, it was not possible to track the progress of an item, since there was no predefined route, but an ad-hoc checking and triggering of new extractors based on MIME-types. Now, with defined workflows (camel routes), progress can be and is tracked. For this purpose, a new progress message for extractors was introduced, which allows extractors to update the broker also on long-running tasks Ÿ OK
- ð FR-7/7a Support for execution planning, semi-automatic route creation: With the initial broker version, planning workflows was not possible, and while the pipeline configuration tools developed and introduced afterwards provided a workaround for this, we learned that a more advanced and usable solution for workflow creation was needed. The

combination of more complex dependency and parametrization descriptions (and the respective, improved broker model), registration service and workflow creation incl. GUI now provide users significant support for route creation and management Ÿ OK

- ð FR-8 Support for dynamic workflow execution: While the initial version of the broker did not support this, the current version now supports two types of dynamic execution based on information derived during analysis workflow: (1) an extractor can produce different types of new parts, depending on its analysis results. The platform then can route the new part to corresponding sub routes. (2) using a combination of Camel, MICO aux components and LDpath, workflows can now include conditional routing based on abstract data elements, which are fetched from the knowledge base (e.g. results from previous extractors) during runtime in order to determine which subroutes are chosen Ÿ OK
- ð FR-9 Type support: All extractors are now using the data model developed in T2.2, and the introduction of Anno4j significantly reduces integration efforts and helps to avoid errors: The new routing includes the validation of types for new parts, before sending them to extractors Ÿ OK
- ð FR-10 Avoid loops: Broker v3 is much less prone to loop issues because the new approach of workflow creation makes them immediately visible, and hence avoids them even before workflows are deployed; beyond that, provenance information can be used by extractors to check whether a part should be analyzed a second time Ÿ OK
- ð FR-11 Intertwined annotation: While this was a requirement with lower priority (currently, no showcase extractor includes human interaction), this could now be easily integrated by implementing a camel component which stops processing of route and waits for human interaction - the design and overall architecture for orchestration supports that (in contrast to broker v1, in which controlled delays could not be distinguished from intended loops etc. Ÿ not tested, but should work
- ð FR-12 Pull content: This was a requirement with low priority, and it is not implemented in the sense of respective API support, because an extractor can do this easily, by querying the knowledge base directly Ÿ superseded
- ð FR-13 Logging: Broker v3 logs provides error and progress tracking, and in case of issues, CATALINA_OUT and extractor logs can be used for tracking issues Ÿ OK; having said that, it could be useful to also include a merged view that does not require privileged access to the server
- ð FR-14 Content-based queries: This feature has not been requested because showcases did not include the need for content-based search and identification, but with broker v3 and pre-defined workflows, it is not difficult to integrate e.g. fingerprint extraction within the analysis workflow, and to perform matching with dedicated identification / CBS services Ÿ not tested, but should work
- ð FR-15 Extractor versioning: All implemented extractors support the semantic versioning concept (see http://semver.org/). This allows broker v3 to differentiate between different extractor versions. The version of an extractor is also stored in the annotation made by this extractor, which means that result comparison among versions is also possible Ÿ OK.

- ð NFR-1 Failure tolerance: With broker v3, extractors can send error information to the broker, and predefined error codes allow the broker to differentiate and handle errors in several ways, e.g. re-queuing for temporal errors, or process cancellation stop for server errors Ÿ OK
- ð NFR-2 Compatibility: Most changes for broker v3 were applied via Messaging API modifications, only minor changes were necessary on extractors itself, and they were introduced early on with the interim version v2, which was based on the broker redesign. As a consequence, respective implementation efforts for extractors could be minimized Ÿ OK
- ð NFR-3 Extensibility: So far, considering the fact that year 3 brought significant changes and improvements regarding persistence and the overall model / Anno4j in parallel that had to be dealt with, the broker design has turned out to be fairly flexible and ñfuture-proofò. The integration of camel for workflow execution also provides the advantage of support for many Enterprise Integration Patterns and protocols that could be used in different setups or for possible extensions Ÿ OK
- ð NFR-4 Testability: The test support of the platform was extended for java extractors. They can be called from a fake platform, to check if they handle incoming request as expected and create correct annotations. For C++ extractors, this functionality could not be implemented anymore. Ÿ only partially implemented. Beyond that, it would also be nice to have support for integrated testing, and possibly even for automatic evaluation or extractors and workflows, all of which was impossible for the project lifetime, but would be great to be addressed in follow-up projects.The good news is that testing both on an extractor and on an integration level have been considered for the design of extractor registration and workflow creation, so all of the aforementioned points could be added without ñarchitectural showstopersò.
- ð NFR-5 Stability: The change from a MIME-Type based routing to dependency checking, pre-defined workflows and Apache camel for execution increased routing performance and reduced routing errors significantly, and the introduction of error messages from extractors allows for more efficient bug tracking and solution. Moreover, the new routing components allow to check types (syntactic- and mime-type) of new parts, before sending them to extractors Ÿ OK; however, as already mentioned for logging, a ñunified viewòon logs would be desirable.

Some further points are worth mentioning in this context:

- ð Broker v3 now supports multiple workflows on the same platform, which was not possible before
- ð Broker v3 is, in principle, able to detect scalability issues and to manage processes based on monitoring the extraction progress - while not implemented, this could be done with a limited amount of efforts, and it would open interesting opportunities re scalability of the extraction process; it is one idea likely to be followed-up on after the project
- ð Broker v3 provides better performance than broker v1 - this is mostly due to the fact that annotation storage and retrieval is faster for pre-defined workflows than it is for the MIME-Type-based approach that resulted in much bigger graphs to be navigated.