EOPEN

opEninterOperable Platform for unified access and analysis of Earth observatioN data

H2020-776019

D4.3

Multimodal fusion for information retrieval

Dissemination level:	Public
Contractual date of delivery:	Month 31, 31/05/2020
Actual date of delivery:	Month 31, 29/05/2020
Workpackage:	WP4 Knowledge discovery and content extraction
Task:	T4.3 Similarity fusion from multiple sources for information retrieval
Type:	Report
Approval Status:	Approved
Version:	1.0
Number of pages:	72

Filename:	D4.3-Multimodal fusion for information retrieval_2020-29-05_v1.0.docx

Abstract

This deliverable reports on methods used for extracting content from EO and non-EO data and for finding similar EO and non-EO content. Specifically, it reports in depth the methodology and the research outputs for finding visually similar EO and non-EO data, and for retrieving data by combining multiple modalities found in each case. Finally, the deliverable contains a methodology for fusing Sentinel (EO data) and social data within the context of snow depth estimation.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 776019

History

Version	Date	Reason	Revised by	Approved By
0.1	02/4/2020	Initial Draft – ToC	Anastasia Moumtzidou (CERTH)	Stefanos Vrochidis (CERTH)
0.2	25/04/2020	Contributions (Section 4)	Damianos F. Mantsis (CERTH)	Ilias Gialampoukidis (CERTH)
0.5	19/05/2020	Contributions (Section 3)	Anastasia Moumtzidou (CERTH)	Stelios Andreadis (CERTH)
0.6	21/05/2020	Final Contributions (Section 2)	Ilias Gialampoukidis (CERTH) Stelios Andreadis (CERTH)	Stefanos Vrochidis (CERTH)
0.7	25/05/2020	Internal Review	Vasileios Sitokonstantinou (NOA)	Gabriella Scarpino (SERCO)
1.0	29/05/2020	Updated document after review for submission	Anastasia Moumtzidou (CERTH) Stelios Andreadis (CERTH) Ilias Gialampoukidis (CERTH)	Stefanos Vrochidis (CERTH) Ioannis Kompatsiaris (CERTH)

Author list

Organization	Name	Contact Information
CERTH	Anastasia Moumtzidou	moumtzid@iti.gr
CERTH	Stelios Andreadis	andreadisst@iti.gr
CERTH	Ilias Gialampoukidis	heliasgj@iti.gr
CERTH	Damianos F. Mantsis	dmantsis@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

This deliverable presents the similarity retrieval and multimodal fusion module that has been developed for handling both EO and non-EO data.

As far as the similarity of non-EO data is concerned, the methodology focuses mainly on data from social media platforms and specifically Twitter data. We start with an in-depth description of the modalities that can be found in Twitter data and provide examples of tweets for all EOPEN use cases. Then, the methods used for single modality search are presented along with state-of-the-art techniques for each case. In the sequel, a late fusion algorithm of multiple modalities is proposed, and the steps of the algorithm are described in detail. Finally, queries are realized for each EOPEN use case in the specific database collections by considering all single modalities, the fusion method and several state-of-the-art fusion techniques. Since the collection of tweets is very big and it is not possible to manually annotate the tweets as relevant or irrelevant, the methodology for evaluating the results is qualitative and is based on the visual inspection of the results for the different methods. The results show that the proposed method produces better results than single modalities and the majority of the existing fusion methods that it is compared against.

Regarding the similarity of EO data, the methodology and the analysis that is conducted focuses on data from satellite images and, specifically, Sentinel data. We begin with an analysis of the multi-label information retrieval task based on similarity that we are about to tackle. Some of the most recent works relative to the fusion of imagery's modalities are presented. Then, we describe the dataset used and the decisions we made to make it suitable for the task. We continue with the description of the modalities that can be found in satellite images, experimenting on both neural network approaches and classic remote sensing techniques. Thereinafter, a late fusion algorithm is used that combines the best performing methodologies of all the available modalities. Image patches are used as queries in order to retrieve the most relevant images from dataset by considering once all single modalities, then the fusion method and finally several other state-of-the-art fusion techniques. Extensive metrics per class for all the methodologies is presented, followed by qualitative analysis that is based on the visual inspection of the top fetched results for the different fusion methods. The quantitative and qualitative results show that the proposed method outperforms single-modality and fusion methods.

The fusion of Sentinel 1 and social media data has also been examined for the estimation of snow depth, where recent studies have shown that it can be estimated accurately on a global scale using satellite images through cross-polarisation and co-polarisation backscatter measurements. However, the existing methods have some limitations in lowland areas with dense forest coverage and shallow snow that is often found nearby urban areas. In these areas, citizen observations can be fused with satellite-based estimations to deliver more accurate solutions. To that end, we use snow-related tweets that have been annotated by artificial intelligence (AI) methods and are introduced in a novel regression model, aiming to increase the estimation accuracy of the state-of-the-art remote sensing method. The proposed model combines the estimated snow depth from Sentinel 1 images with the number of Twitter posts and Twitter images that are semantically relevant to snow. The use of social media data for purposes of snow depth estimation is investigated, validated and tested in Finland. Our results show that this approach does improve the snow depth

estimation, highlighting its potential for use in civil protection agencies in managing snow conditions, by fusing Sentinel 1 images and social data.

Abbreviations and Acronyms

ADC	Asymmetric Distance Computation
AP	Average Precision
API	Application Programming Interface
BERT	Bidirectional Encoder Representation
BOW	Bag Of Words
BR	Backscatter Radiation
CBOW	Continuous Bag-of-Words
DCNN	Deep Convolutional Neural Network
CMC	Canadian Meteorological Centre
DNN	Deep Neural Network
CSN	Classification Similarity Networks
EO	Earth Observation
FC	Fully Connected
GeoJSON	Geographic JSON
JSON	JavaScript Object Notation
FoC	Forest Cover
mAP	Mean Average Precision
MSE	Mean Squared Error
NN	Neural Network
PUC	Pilot Use Case
NIR	Near-Infrared
SAR	Synthetic Aperture Radar
SC	Snow Cover
SD	Snow Depth
SQL	Structured Query Language
SWIR	Short-Wavelength Infrared
TF	Term Frequency model
TFIDF	Term Frequency Inversed Document Frequency
VSM	Vector Space Model

Table of Contents

1 INTRODUCTION

The amount of space-borne Earth observation data that is obtained increases day by day due to the multitude of sources orbiting around the globe. This advance of the satellite remote sensing technology produces the necessity of quick and precise generation of land cover maps that distinguish the characteristics of the underlying areas, providing beneficial information to global monitoring studies, resource management, and planning activities. To address this challenge, information retrieval undertakes to extract the attributes that characterize some satellite imagery in order to retrieve the most closely related images. Many characteristics can be used, varying from the visual content and the general concept that the areas depict, or even the location and the time of the data acquisition. Combining them together increase the possibilities to find areas with the same characteristics.

Within this context, EOPEN extracts knowledge from the collected EO and non-EO data in order to add value in data related to flood monitoring, food security and extreme weather conditions. This document presents the developed information retrieval techniques that consider either a single modality or fuse multiple modalities and which are applied both on EO and non-EO data. Furthermore, it presents a technique for combining EO with non-EO data for snow depth estimation.

In Section 2, we discuss the techniques for similarity retrieval for non-EO data, i.e. Twitter data. The techniques presented are single modality data retrieval techniques that differ per type of modality and a new multiple modalities data retrieval method that is basically a late fusion algorithm that considers the output of the single modalities. The results of the methods are discussed and compared against the ones of other well-known late fusion methods.

In Section 3, we analyse various similarity retrieval techniques focusing on EO data and more specifically on optical satellite imagery. The methodologies presented are single modality data retrieval techniques that differ per type of modality. A variety of both deep learning and classic remote sensing methods are explored. Eventually, a new multiple modalities data retrieval method is evaluated. The results of the method are compared against the ones of other well-known late fusion methods.

Section 4 focuses on using cross-polarisation and co-polarisation backscatter satellite measurements to estimate snow depth in Southern Finland. Due to the limitations of this method in lowland areas with dense forest coverage and shallow snow, we fuse the remote sensing data with citizen observations (snow-related Twitter posts which are annotated by deep learning methods) through a regression model, aiming to increase the estimation accuracy of the state-of-the-art remote sensing method.

Finally, Section 5 concludes the deliverable and discusses the main findings produced from each section.

2 DATA FUSION FOR NON-EO CONTENT FOR INFORMATION RETRIEVAL

The need to retrieve similar visual content from a set of observations in response to a query will be tackled in this task. Each item in the collection is equipped with several modalities (e.g. visual, textual, and spatiotemporal) that need to be fused in a scalable way, taking into account memory and computational complexity, in order to retrieve similar content. The output of this task is the EOPEN search engine, which will provide the top-k related EO products or social media posts, in response to multimodal query. Social media posts and EO imagery are associated and linked with metadata (tags, extracted concepts, text, time, location), but matching the similarities per modality for all modalities is not a scalable solution. The EOPEN fusion of similarities is based on the unsupervised fusion of similarities (Ah-Pine et al., 2015), which has been extended to multiple modalities (Gialampoukidis et al., 2016a), under the same memory complexity. This task will start with the development of an annotated dataset, for training purposes, in order to tune the parameters of the model in the context of the EOPEN use case scenarios. Tuning will be followed by an evaluation in the significance of each involved parameter and modality, and several directions towards the model simplification will be examined. The output of this task is a module, able to compare two multimodal objects, integrating all sources of information, effectively and quickly.

This section tackles the fusion and retrieval of multimodal non-EO content collected within EOPEN. As non-EO content, we consider data collected from social media platforms and in particular data from Twitter. While, the collection of data will be described in detail in EOPEN deliverable D3.3 "EOPEN Social Media Crawlers"**,** this section focuses on the techniques applied for the efficient retrieval of such content in response to a query tweet. It should be noted that each tweet item is equipped with several modalities, including visual, textual, and spatiotemporal. The aim of this task is to consider all the aforementioned modalities, fuse them in a scalable way, taking into account memory and computational complexity, in order to retrieve similar content in real-time (i.e. 1-10 seconds maximum retrieval time) using AI.

2.1 Methodology

This section provides a detailed description of the approach followed for retrieving similar tweets. In order to make clear the reasons that led to the selection of the proposed approach, it is necessary to describe adequately the information linked to each tweet. Thus, each tweet contains the following information:

- A short text no longer that 140 characters that may contain non-standard terms, misspellings, "emojis", slang and abbreviations
- Possibly an image that is usually semantically related to the text
- The date and the time of the tweet publication

Figure 1 depicts examples of 3 tweets, one for each EOPEN Use Case, i.e. Floods, Food and Snow. The languages of the tweets are Italian for the Flood use case and Finnish for the Snow use cases, according to the country each Pilot focuses on. However, for the Food Security Use Case, the English language is opted instead of Korean because analysing ideograms is beyond the scope of the project.

Tweet id: 1213451042873995264

PUC: Flood **Language**: Italian

Text:
Ferrovie, riparte la Bologna-Portomaggiore interrotta dall'esondazione dell'Idice https://t.co/OjBZNcHjh1 via @user https://t.co/ZRY5DTlR14 posted by GKOhefs

Date / Time:
Saturday 04 January 2020, 15:24

Tweet id: 1116207503442894849

PUC: Food **Language**: English

Text:
👩‍🌾🌾👨‍🌾🌾Farmers are our more than just food producers, they are guardians of our natural resources and born innovators!
Empowering our farmers and investing in innovation is key to achieving #ZeroHunger https://t.co/ZpFq6BILAN #AgInnovation https://t.co/on8aFc9hnk

Date / Time:
Thursday 11 April 2019, 08:13

Tweet id: 1251016748649844737

PUC: Snow **Language**: Finnish

Text:
Aamuhankia pitkin Kuninkaanlaavulle. Komeat on maisemat Ounasjoelle vaikka tuuli tuiversi ja lunta tuiskusi. Ei ollut kovin keväistä, mutta ehkä se talvi täältäkin joskus loppuu. #Kuninkaanlaavu #Santavaara #Rovaniemi #hankikanto https://t.co/dPtXndro6Y
posted by xd++EZg

Date / Time:
Friday 17 April 2020, 08:16

Figure 1: Examples of 3 tweets, one for each EOPEN Use Case, i.e. flood monitoring, food security and snow observations.

Therefore, the retrieval of similar social media posts revolves around this information and any other information extracted by it after applying processing and analysis techniques.

Once a Twitter post is given as a query, several modalities are involved such as textual, visual, spatial and temporal information. Starting from the text information, textual representation of the tweet can be applied that will allow retrieval of similar text. Furthermore, after applying named entity recognition technique in the tweet text, locations and organization mentioned to the tweet can be identified which can be linked to a specific geo location and eventually allow the retrieval of geographically close tweets. As far as the image information is concerned, images can be described both with low level features and high-level visual concepts, thus allowing the retrieval of visually similar images to the image of the query and retrieval of images described with similar visual concepts respectively. Finally, temporal information can be used for retrieved tweets that are close in terms of date and timestamp.

Figure 2 depicts all the metadata that are produced after applying the techniques mentioned in the following section in order to describe a tweet, i.e. vector of visual concepts, text, time, location and visual similarity.

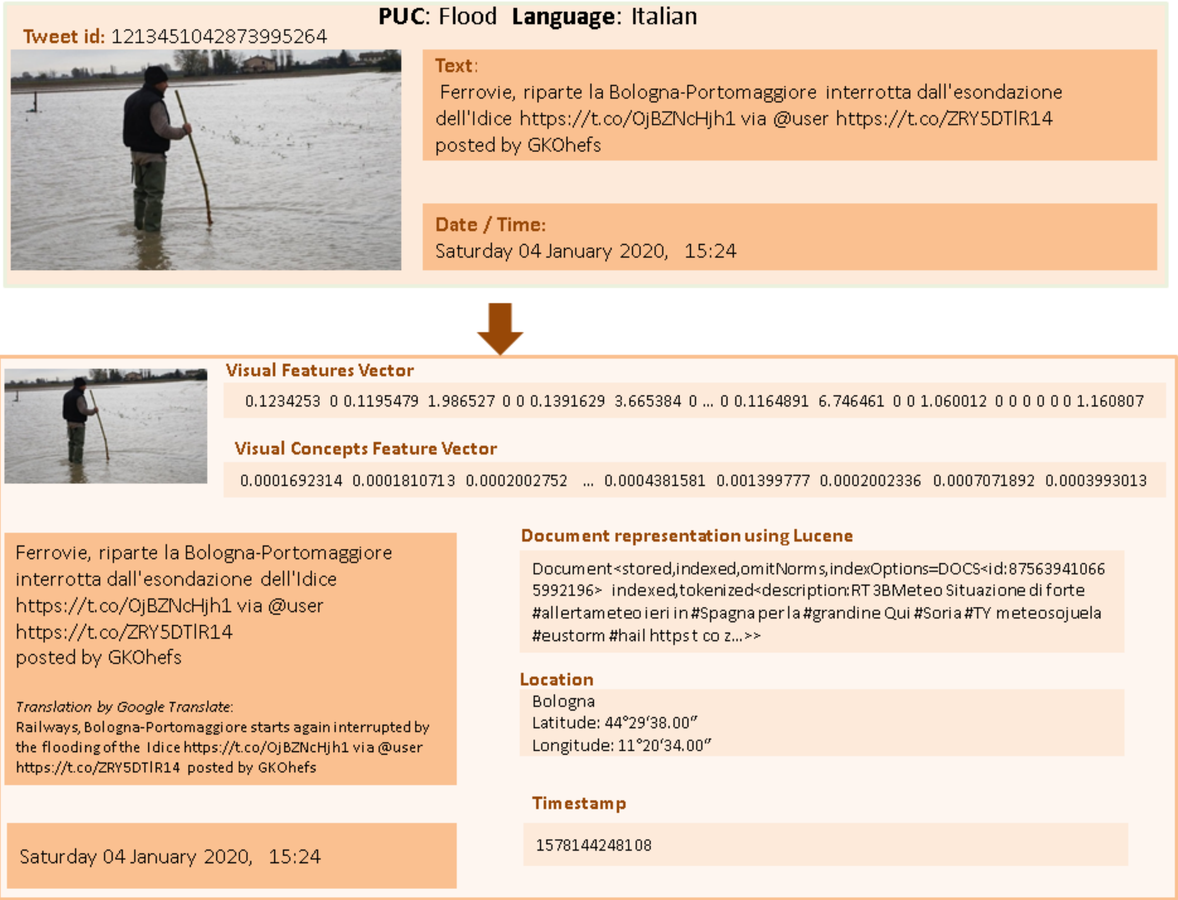

Figure 2: Tweet and metadata produced after applying processing techniques.

Given the heterogeneity of the metadata produced that describe a tweet from different perspectives, a late fusion approach was considered as a sole solution.

In the following, the methods for obtaining the metadata from the tweet and the fusion methods used are described in detail.

2.1.1 Similarity by textual content

Text similarity between two or more texts is the procedure of computing the similarity in meanings between them. There are several approaches that can be used for text similarity that involve as a first step, text representation, then as a second the distance function to calculate the distance among different texts. Finally, the distance measures of the texts are ranked from lowest to higher and the ranked set of documents is the output of the similarity by textual context module. In the following we describe some text representation techniques and distance functions used.

As far as text representation (Yan, 2009), is concerned the most commonly used text representation model is Vector Space Model (VSM) where documents are represented by vectors of words and a typical VSM is the Bag of Words (BOW) which uses all words of a given document set D as the index of the document vectors. Several term weighting schemas exist under the BOW model, including the Boolean model which involves the binary representation of documents, the Term Frequency model (TF) that uses the frequency of the terms, and the Term Frequency Inversed Document Frequency (TFIDF) model that considers real values that capture the term distribution among documents to weight terms in each document vector. A more recent approach that generally outperforms the other methods in many cases, is word2vec (Mikolov, et al. 2013). word2Vec algorithm is a model that produces word embeddings (i.e. representation of words from a given vocabulary as vectors in a low-dimensional space) and builds distributed semantic representation of words, based on deep neural networks (NN), which are either the Continuous Bag-of-Words model (CBOW) or the Skip-gram. Both models are trained on large corpus, taking into consideration the neighbouring words in a sentence. The difference between these two architectures is that while in the CBOW the NN model tries to predict a word given the context of this word, in the Skip-gram given a word the NN model tries to predict the context of a word. The same idea of word2vec can be extended to sentences and documents where instead of learning feature representations for words, what the model learns is sentences (SentenceToVec) or documents (Doc2Vec) and that can be considered as a mathematical average of the word vector representations of all the words in the sentence. Another approach similar to word2vec is GloVe (Pennington, et al. 2014) which is an unsupervised learning algorithm for obtaining vector representations for words and thus no model is required. In GloVe, training is performed on aggregated global word-word co-occurrence statistics from a corpus. Finally, another more recent approach is the Bidirectional Encoder Representation from Transformers (BERT) algorithm (Devlin, et al. 2018). BERT is a non-directional or bidirectional model that involves an attention mechanism that learns contextual relations between words in a text and reads the entire sequence of words at once.

In order to calculate the similarity between two sequences of strings there are a number of string similarity measures. Some of the most popular term-based distance measures are the Manhattan distance, the cosine similarity, the Dice's coefficient, the Euclidean distance, the Jaccard Similarity, the Overlap coefficient and the Matching coefficient (Vijaymeena, 2016).

Apart from the aforementioned methods, there are some off-the-shelf text search engines with most prominent one, the Apache Lucene. Apache Lucene[1] is a full-text search engine which can be used from various programming languages and can be used for any application that that requires full text indexing and searching capability. Lucene is recognized for its utility in the implementation of Internet search engines and local, single-site searching. A list of the companies that use Lucene for their product or website can be is maintained by the Lucene team and it can be found **here**. Among the biggest deployments are Twitter that uses Lucene to power its real-time search over tweets, which is over a billion queries a day, LinkedIn which has also modified and enhanced Lucene for real-time search and faceted search, Hi5 and Comcast.

Lucene is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead, which can be considered equivalent to a glossary at the end of any book. This type of index is called an inverted index, because it inverts a page-centric data structure (page->words) to a keyword-centric data structure (word->pages).

Indices consist of one or more documents, and search results are sets of best-matching documents. A document is a collection of fields, and each field has a value associated with it.

[1] https://lucene.apache.org/

This value is typically text which is converted into smaller and precise units during an analysis step in order to allow easy search. Specifically, the text goes through various operations which include extracting keywords, removing common words and punctuations, changing words to lower case, etc. For this purpose, there are multiple built-in analyzers:

- **StandardAnalyzer**: analyses based on basic grammar, removes stop words like “a”, “an”, lowercases the token and in general is the most sophisticated analyser
- **SimpleAnalyzer**: breaks the text based on no-letter character and converts in lowercase
- **WhiteSpaceAnalyzer**: breaks the text based on white spaces

It should be noted that there are Analyzers used that are dependent on language.

Finally, once an index is built, it is possible to search the created index using a Query and an IndexSearcher. The search result is typically a result set, containing the retrieved data. Finally, Lucene provides a very dynamic and easy to write query syntax that allows the user to specify which field(s) to search on, which fields to give more weight to (boosting) and also the ability to perform Boolean queries.

After this brief overview of the available solutions for text similarity, we opted for Apache Lucene within the context of EOPEN. The reason is that although the aforementioned techniques (e.g. BERT) may be more efficient in terms of quality of results, Apache Lucene allows very fast indexing and retrieval, which is of critical importance in the case in EOPEN where tweets are retrieved every second and thus the size of the collection is expanding extremely fast. In the Big Data context of EOPEN, it is necessary to have an index that can be updated very fast and also that allows fast and efficient retrieval over more than 10,000,000 tweets. Under the light of this requirements set by the EOPEN and after checking carefully all the possible solutions we concluded that Apache Lucene is more fitted. Furthermore, we should note that EOPEN we have three different languages, i.e. Italian, Finnish and Korean for each use case and English for easy and widespread demonstration of the platform. Thus, we considered the following language-specific analysers in order to handle efficiently the three aforementioned languages:

- org.apache.lucene.analysis.en.EnglishAnalyzer
- org.apache.lucene.analysis.fi.FinnishAnalyzer
- org.apache.lucene.analysis.it.ItalianAnalyzer
- org.apache.lucene.analysis.ko.KoreanAnalyzer

Finally, we should note that different Indexes were created for each pilot case and each language, that are updated whenever a new tweet is available, resulting in the six following Indexes:

- ItalianFloods
- EnglishFloods
- KoreanFood
- EnglishFood
- FinnishSnow
- EnglishSnow

2.1.2 Similarity using visual information

As far as similarity by visual information is concerned, it involves similarity by visual content and similarity by visual concepts. The framework used in both cases, i.e. a deep neural network, is the same, but the vectors used are taken from different layers of the network. Regarding the description of State-of-the-Art techniques, they were already provided in EOPEN deliverable D4.1 deliverable, entitled "Change detection techniques in Earth Observation", in Section 3.1.1 ("State of the art in Concept Detection") and in Section 4.1 ("State of the art in Similarity Fusion").

In the following, we will briefly describe the deep neural network used, as it was also described in D4.1. A 22-layer GoogleNet network (Szegedy, 2015) was trained on 5055 ImageNet concepts (Pittaras, et al. 2017), which are a subset of the ImageNet "fall" 2011 dataset[2] that was trained originally on 32,326 concepts. The subset of the 5055 concepts was produced by merging very similar concepts, removing concepts corresponding to scientific terms, and concepts with a very few number of positive images. Therefore, the dimension of classification layer of the trained network, which is a fully connected layer, equals to 5055. Following the GoogleNet architecture, Pittaras (2017) added after the classification layer a *softmax* function. The number of concepts identified was reduced even more in order to target the TRECVID Semantic Indexing SIN 2013 task[3], and thus the authors ended up with 345 SIN TRECVID concepts[4]. In order to train these new concepts, fine-tuning was performed and after evaluating different fine-tuning methods, the one that performed the best involved replacing the classification layer with dimensionality 5055 with a classification layer with dimension equal to 345. It should be noted that GoogleNet has by default three classification layers. Thus, in order to keep the GoogleNet architecture, the authors considered three classification layers with dimension equal to 345. Finally, based on research realized on fine-tuning (Pittaras, 2017), an extra fully connected layer was added right before the classification layers, as it seems to boost its performance. Figure 3 depicts the original GoogleNet architecture and the described fine-tuned GoogleNet.

[2] http://academictorrents.com/details/564a77c1e1119da199ff32622a1609431b9f1c47

[3] https://www-nlpir.nist.gov/projects/tv2013/index.html

[4] http://www-nlpir.nist.gov/projects/tv2012/tv11.sin.500.concepts_ann_v2.xls

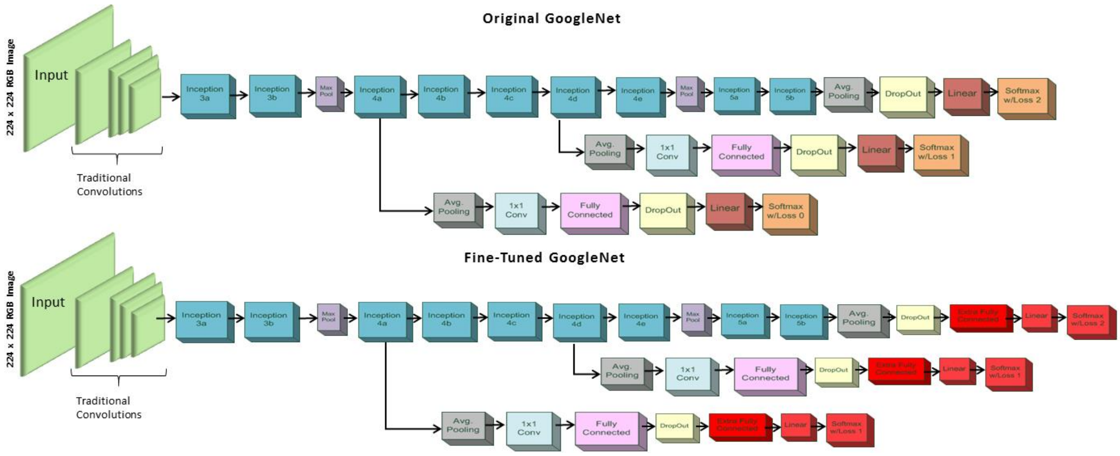

Figure 3: Original GoogleNet architecture and fine-tuned GoogleNet (layers in red have been added or replaced layers existing in the original GoogleNet).

Similarity by visual content

Regarding the visual features, they are DCNN-based descriptors and they are the output of the last pooling layer of the fine-tuned GoogleNet architecture (Pittaras, 2017) previously described. The dimension of the last pooling layer is 1024 and it is used as global image representation. The selection of a DCNN-based feature was based on the outcome of several studies that revealed the superiority of such features versus hand-crafted features both in terms of accuracy and time. Furthermore, as far as the selection of the last pooling layer for representing the image is concerned, it was evaluated both in terms of time and quality of results within the VERGE system (Moumtzidou, et al. 2018) that has participated in the Video Browser Showdown[5] in 2018. The dataset, it was evaluated on, was the IACC.3 dataset[6] that was used on the TRECVID 2018 AVS Task[7] and which consists approximately of 4600 Internet Archive videos (144 GB, 600 h) with Creative Commons licenses with duration ranging from 6.5 min to 9.5 min and a mean duration of almost 7.8 min.

Figure 4 shows the layer of the GoogleNet architecture that is used as DCNN-feature.

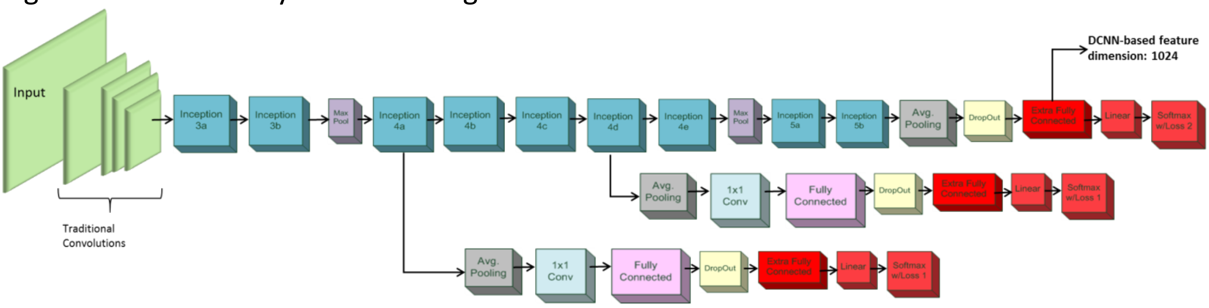

Figure 4: Extraction of DCNN-based feature from the fine-tuned GoogleNet.

In order to retrieve visually similar images fast and efficient, we followed the Nearest Neighbour search which is the best performing approach between the query and database vectors described in (Jegou, et al. 2010; Jegou, et al. 2011) is applied. This approach involves initially, the construction of an inverted file and then combining it with Asymmetric Distance

5 https://videobrowsershowdown.org/
6 https://www-nlpir.nist.gov/projects/tv2018/data/#IACC.3
7 https://www-nlpir.nist.gov/projects/tv2018/Tasks/ad-hoc/

Computation (ADC). Even though the existence of such an index, that is produced from the feature vector of the images and a unique identifier per image (i.e. tweet Id) speeds up significantly the querying time, the loading of the index to the RAM requires significant time, which might rise up to several minutes in large databases (i.e. number of records greater than 500,000) which is the case for several of the databases in EOPEN. Therefore, in order to eliminate the time required for the index loading, a simple REST web service is created that loads permanently this indexing structure in RAM, and also allows querying the index. By using this procedure, instant querying of the structure and eventually fast results retrieval is achieved each time a visual query is realized.

Finally, it should be noted that the indexing structure is updated on a daily basis since it is not time-efficient to update the index every time a new image is available. This is due to the fact that the time needed for reloading the index in the memory takes us to 5 minutes.

Similarity by visual concepts

Regarding the visual concepts, they are the output of the fine-tuned GoogleNet architecture (Pittaras, 2017) previously described. Thus, we have available the probabilities of 345 concepts. These probabilities are concatenated to a single vector with length 345, which is used for capturing the concepts found in each image.

Figure 3 shows the layer of the GoogleNet architecture that is used as concept vector. Similarly in Figure 4, the neural network layer that is used to extract the visual feature vector is presented. In order to retrieve visually similar images in a fast and efficient way, we followed the same indexing Nearest Neighbour search in both visual feature and visual concept search. Thus, an indexing is created that uses the concept vectors and a unique identifier per image (i.e. tweet Id) and then a simple REST service is created that loads the index to the RAM and also accepts requests.

Following the same pattern as before, the indexing structure is updated on a daily basis since it is not time-efficient to update the index every time a new image is available. This is due to the fact that the time needed for reloading the index in the memory takes about to 5 minutes (per day).

Figure 5 depicts the procedure described, starting from the generation of the feature vector or the concept vector, the query to the index and finally the retrieval of the results.

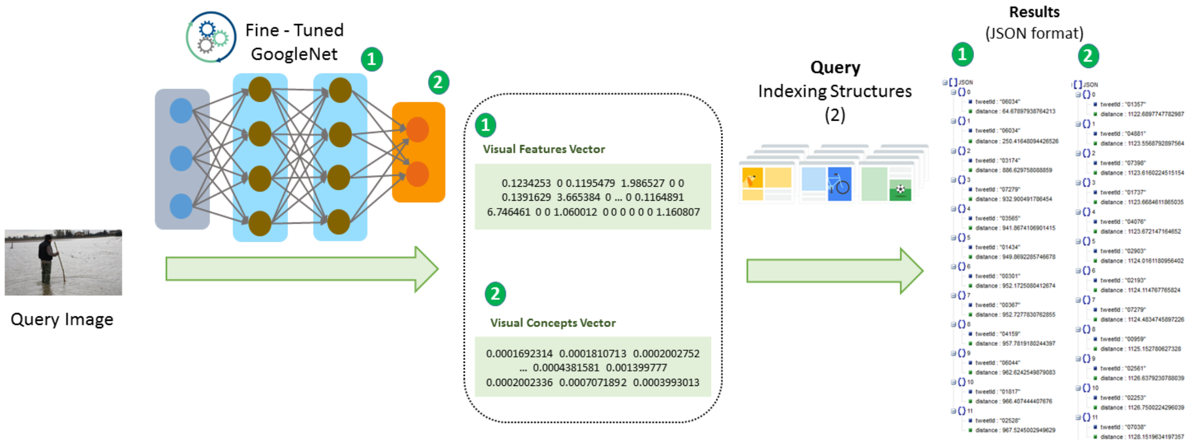

Figure 5: Visual Information Generation and retrieval procedure.

2.1.3 Similarity by time and geolocation metadata

Temporal information

The similarity considering time and geolocation information is tackled in a different manner compared to the other information. At this point, it should be noted that all the information that is related to the tweets and is either directly taken from the Twitter or is produced after analysing them is stored to a non-SQL database, and specifically MongoDB[8]. The exact information that is stored per tweet will be described in the upcoming deliverable D3.3 ("EOPEN Social Media Crawlers"). However, we should note that for each tweet, apart from the text, that was described in Section 2.1.1 , the time of the publication is provided as well. This information is kept also in timestamp format thus allowing easy sorting using MongoDB.

Below there is an example of a date/time converted to a timestamp in millisecond.

```
05/05/2020018:06:56 ⇨ 1588691216000
```

Thus, by considering the timestamp of every tweet, we used the absolute value of the difference between the timestamp of the tweet query and the timestamp of each tweet in the database to sort the tweets. Then, we sorted these values in increasing order and kept the *N* smaller values which stand for the tweets that are closer to the tweet query in terms of its timestamp.

Spatial information

Similar to the time information, the spatial (geolocation) information is stored in the MongoDB is well. However, Twitter does not provide geographical information for the majority of tweets, because the twitter users do not enable location information as part of their Twitter post. To that end, moreover, the meaningful location that needs to be analysed is the location entity that appears in the text and not the place in which the tweet is posted on Twitter. The spatial information from Twitter content is produced as part of analysis realized on the Twitter's text.

Specifically, a named entity recognition (NER) method is employed in order to locate organization and location entities found in user tweets, which are then pinpointed to a map via the OpenStreetMap API. Currently, the deep neural networks-based approach exploits a bidirectional LSTM-CRF model (see D5.1 "The EOPEN ontology and semantic reasoning support", section 7.1) which will be updated with ELMO-based embeddings in the upcoming deliverable D5.2 ("Semantic reasoning for decision making"). Then, this information is stored as a geospatial data, and specifically a GeoJSON Point, using GeoJSON objects to the MongoDB.

The GeoJSON, as defined within MongoDB, has the following structure:

- a field named type that specifies the GeoJSON object type and
- a field named coordinates that specifies the object's coordinates
- latitude and longitude coordinates, which includes listing first the longitude and then the latitude. The types of GeoJSON objects that are supported from MongoDB are: Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultiPolygons

[8] https://www.mongodb.com/

```
<field>: { type: <GeoJSON type> , coordinates: <coordinates> }
```

Figure 6 depicts an example of the localisation procedure involving all the aforementioned steps.

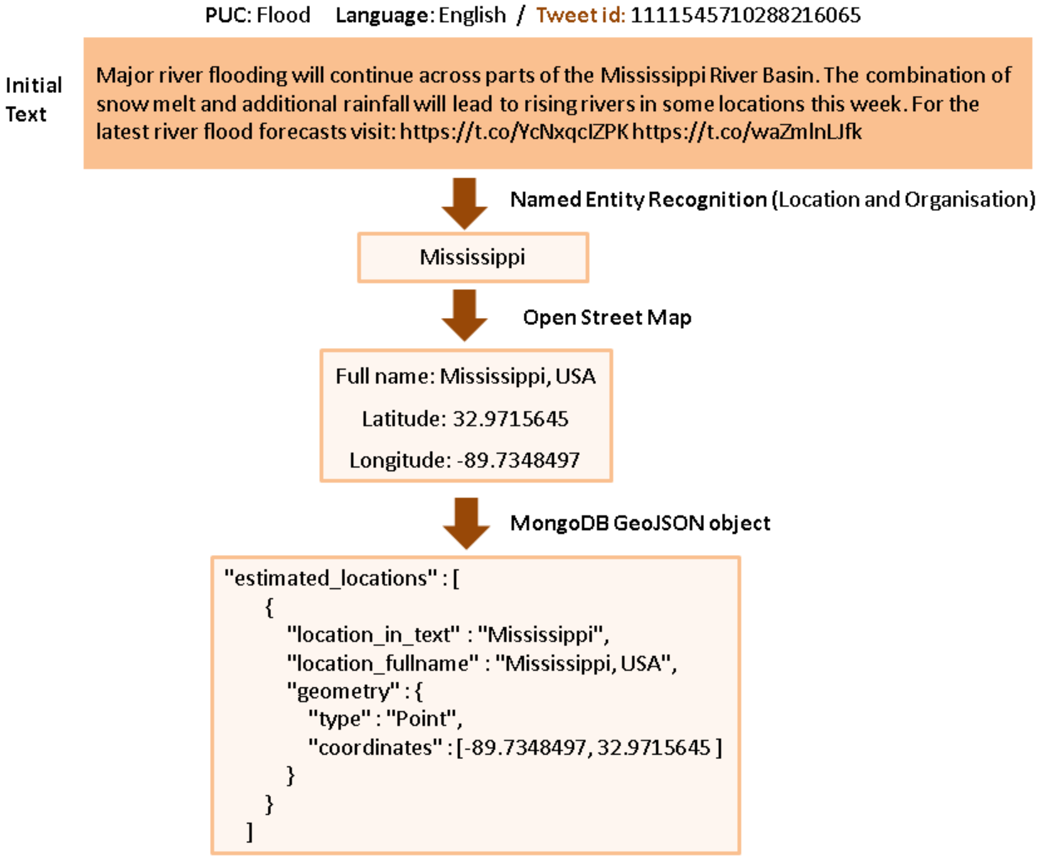

Figure 6: Overview of geolocation information extraction.

In order to obtain the tweets that are close in terms of geolocation, a built-in function of MongoDB is used that considers the information stored in the GeoJSON objects. Specifically, we consider *$geoNear*[9] that returns documents in order of nearest to farthest from a specified point. Some important parameters that should be considered: a) the distanceField option should be used, and b) a geospatial index must exist.

Figure 7 depicts an example of the query send to MongoDB in order to retrieve the documents that are closer in terms of geolocation:

[9] https://docs.mongodb.com/manual/reference/operator/aggregation/geoNear/

```
collection.aggregate([
  {
    '$geoNear': {
      'near': {
        'type': "Point",
        'coordinates': [
          long,
          lat
        ]
      },
      'distanceField': "dist.calculated",
      'spherical': True
    }
  }
])
```

Figure 7: Example of *$geoNear* MongoDB query

2.1.4 Fusion of modalities

This section describes the framework used for fusing the aforementioned information, textual, visual features, visual concepts, time and geolocation. An overview of the state of arts for similarity fusion was provided in EOPEN deliverable D4.1 "Change detection techniques in Earth Observation" in section 4.1 - State of the art in Similarity Fusion.

Due to the heterogeneity of the information/modalities used for describing a tweet, a late fusion approach was opted. In the following we describe in detail the proposed algorithm.

Algorithm and Notation

The algorithm proposed can be applied for fusing the output of K modalities, where $K \geq 2$. For each modality, we have N retrieved results and thus we have K such lists. We set as $\underline{\boldsymbol{L}}$ the K-order tensor of the retrieved lists, L_θ, $1 \leq \theta \leq \mathrm{K}$. A single element $\underline{\boldsymbol{l}}$ of $\underline{\boldsymbol{L}}$ is addressed by providing its exact position through a series of indices $r_1, r_2, \ldots, r_K$ i.e.:

- $\underline{\boldsymbol{l}}_{r_1,r_2,\ldots,r_K} \equiv \underline{\boldsymbol{L}}_{r_1,r_2,\ldots,r_K}; 1 \leq r_\theta \leq N$
- $\underline{\boldsymbol{l}}_{r_1,r_2,\ldots,r_K} = 1$ if the same element w_n, $1 \leq n \leq N$ (e.g. Twitter ID) has rank r_1 in list L_1, rank r_2 in list L_2, …, and rank r_K in list L_K.

The aim of the algorithm is to find the final list L^f of retrieved results. Thus, the first step of the algorithm is to compute tensor $\underline{\boldsymbol{L}}$ and the second is the computation of the final list L^f.

Algorithm 1: Compute tensor $\underline{\boldsymbol{L}}$

Input: L_θ, $1 \leq \theta \leq \mathrm{K}$

for each element w_n, $1 \leq n \leq N$

  if $rank(w_n) = \{r_1 \in L_1\} \wedge \{r_2 \in L_2\} \wedge \cdots \wedge \{r_K \in L_K\}$:

    $\underline{\boldsymbol{l}}_{r_1,r_2,\ldots,r_K} = 1$

  else:

    $\underline{\boldsymbol{l}}_{r_1,r_2,\ldots,r_K} = 0$

Output: $\underline{\boldsymbol{L}}$

Algorithm 2: Get the final list L^f

Input: $\underline{\boldsymbol{L}}$

$j = 1$

while $j \leq N$

$L^f = \emptyset$

if $\left\|\underline{\boldsymbol{L}}_{r_1 \leq j, r_2 \leq j, \ldots, r_K \leq j}\right\|_2 > 0$

which w_n s.t. $\underline{\boldsymbol{l}}_{r_1 \leq j, r_2 \leq j, \ldots, r_K \leq j} = 1 \; (1 \leq n \leq N)$

$L^f = L^f \cup \{w_n\}$

else:

$L^f = L^f$

$j = j + 1$

Output: L^f

Example of the Algorithm

In order to understand better how this algorithm, an example is provided. Thus, we consider that we have $\mathrm{K} = 4$ modalities, i.e. text, time, location, and visual features, and that the number of retrieved results per modality is $\mathrm{N} = 5$. Figure 8 depicts the lists with the retrieved results per modality.

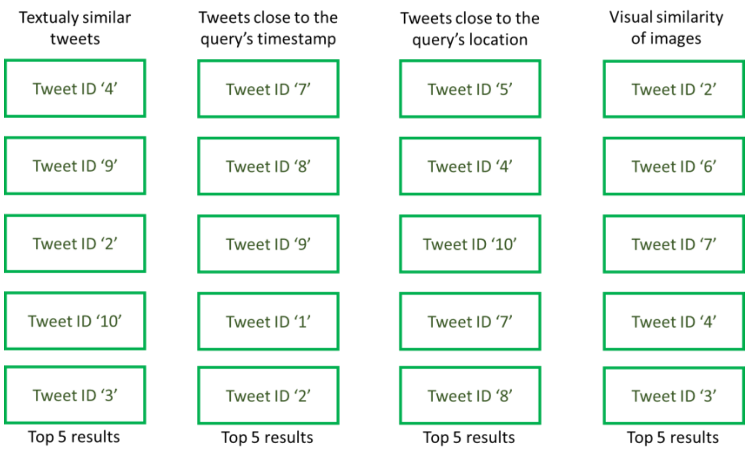

Figure 8: Lists with the retrieved results per modality

Based on the list of retrieved results, we fill in the surfaces of the $\underline{\boldsymbol{L}}$ tensor as shown in Figure 9.

Text / Time

	‘7’	‘8’	‘5’	‘1’	‘2’
‘4’	0	0	0	0	0
‘9’	0	0	0	0	0
‘2’	0	0	0	0	1
‘10’	0	0	0	0	0
‘3’	0	0	0	0	0

Text / Location

	‘5’	‘4’	‘10’	‘7’	‘8’
‘4’	0	1	0	0	0
‘9’	0	0	0	0	0
‘2’	0	0	0	0	0
‘10’	0	0	1	0	0
‘3’	0	0	0	0	0

Text /Visual Features

	‘2’	‘6’	‘7’	‘4’	‘3’
‘4’	0	0	0	1	0
‘9’	0	0	0	0	0
‘2’	1	0	0	0	0
‘10’	0	0	0	0	0
‘3’	0	0	0	0	1

Time/ Location

	‘5’	‘4’	‘10’	‘7’	‘8’
‘7’	0	0	0	1	0
‘8’	0	0	0	0	1
‘5’	1	0	0	0	0
‘1’	0	0	0	0	0
‘2’	0	0	0	0	0

Time/ Visual Features

	‘2’	‘6’	‘7’	‘4’	‘3’
‘7’	0	0	1	0	0
‘8’	0	0	0	0	0
‘5’	0	0	0	0	0
‘1’	0	0	0	0	0
‘2’	1	0	0	0	0

Time/ Visual Features

	‘2’	‘6’	‘7’	‘4’	‘3’
‘5’	0	0	0	0	0
‘4’	0	0	0	1	0
‘10’	0	0	0	0	0
‘7’	0	0	1	0	0
‘8’	0	0	0	0	0

Figure 9: Example of surfaces of $\underline{\boldsymbol{L}}$ tensor (result of 1st Algorithm)

Finally, **Error! Reference source not found.** depicts the implementation of the 2nd algorithm, which involves five steps till the final result. Specifically the first step is the bi-modal fusion of the retrieved results, the second is the bi-modal ranking of the retrieved results, the third involves the merging of the rankings, the fourth the duplicate removal and finally the fifth includes getting the final list $\boldsymbol{L}^f$.

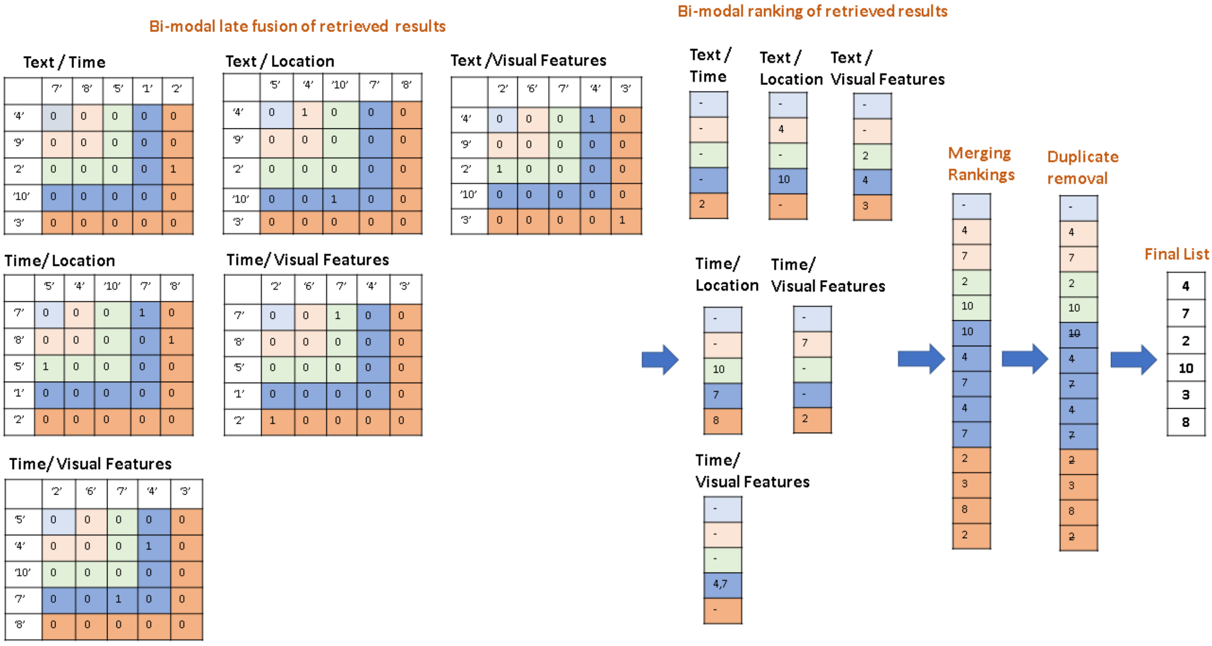

Figure 10: Steps of ranking procedure described in 2nd Algorithm.

2.2 Results and Discussion

This section presents the results retrieved from each single modality, and from the fusion of the modalities using the approach described in Section 2.1.4 and some well-known State-of-the-Art fusion methods. Specifically, the late fusion against which we will evaluate the proposed method are the seminal work of Borda fusion (Borda, 1784), Condorcet fusion (Montague and Aslam, 2002) and Reciprocal rank fusion (Cormack, et al. 2009). It should be noted that in order to evaluate the results of the different methods, we consider a qualitative method for the non-EO data, since it is not feasible to annotate the collected tweets collected. This involves the visual inspection of results and commenting on them. Regarding the quantitative results, we check and manually annotate, by considering the text of the tweet, whether a tweet is relevant or not to floods, food or snow, on the *top-N* retrieved results for each retrieval method and calculate the average precision for each query and mean average precision for 3 queries for each method. Specifically, for each different retrieval method, we will evaluate the *top-10* retrieved results.

As already mentioned, the tweets gathered are in 4 languages, i.e. Italian, Finnish, Korean and English and they cover different use cases (i.e. flood, food, snow). Thus, as defined in the Grant Agreement, the flood use case refers to the area of Italy, the snow case to the area of Finland and the food case to the area of Korea. However, as already mentioned analysing ideograms is a difficult task and beyond the scope of the project, we will consider English tweets for the case of food. Therefore, we will evaluate the results from the following queries:

- 1 query tweet in Italian language that is related to flood use case (Figure 11)
- 1 query tweet in Finnish language that is related to snow use case (Figure 12)
- 1 query tweet in English language that is related to food use case (Figure 13)

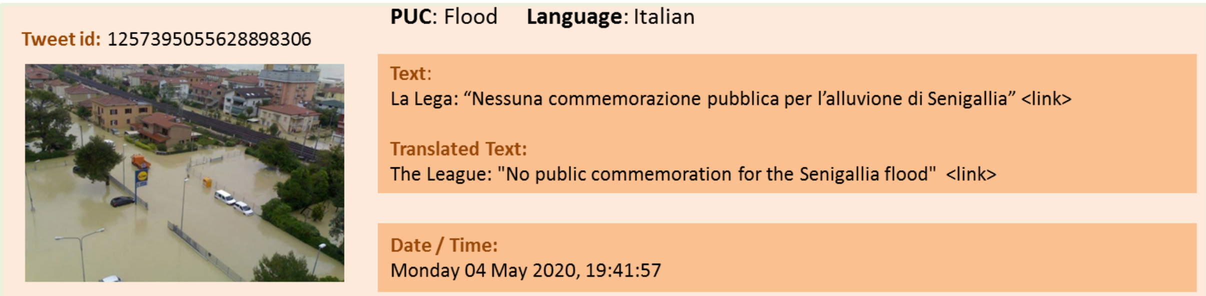

Figure 11: Query tweet in Italian language that is related to flood use case.

Figure 12: Query tweet in Finnish language that is related to snow use case.

Figure 13: Query tweet in English language that is related to food use case.

Table 1 contains the average precision scores for the different similarity methods for each query and the mean average precision (mAP) for each method. Moreover, in Appendix A.1, there are screenshots of the *top-10* tweets retrieved for all methods and for the 3 query tweets. In general, we can draw the following conclusions:

- The text modality has the lowest score when not fused with additional information. However, it is the only modality together with temporal, that exists in each tweet.
- The fact that time modality has better mAP compared to text is due to the fact that we retrieve the top-10 results only and thus it is more probable that tweets that are near in terms of time and have similar keywords to be related. However, it is expected that if we retrieve more results, this score (mAP) will fall.
- Visual features have very good mAP since it searches for visually similar results and isn't based on models (such as Visual Concepts) whose performance depends highly on how good the training set is.
- From the fusion techniques the ones that perform the best are the proposed EOPEN algorithm and the Borda fusion algorithm. However, the irrelevant retrieved results are ranked higher in Borda fusion algorithm, a fact which also affects the performance of a search engine. Qualitative analysis (Appendix A) with visual inspection on more than the top-10 retrieved results show superiority of our proposed method, when compared to single and multiple modality fusion methods.

Table 1: Average precision and mean Average Precision

		Average Precision@10			mean Average Precision
		Flood (IT)	Food (EN)	Snow (FI)	
Single modalities retrieval	**Text**	1.0	1.0	0.586	0.862
	Time	0.839	0.867	1.0	0.902
	Visual Features	0.878	1.0	1.0	0.959
	Visual Concepts	0.638	1.0	1.0	0.879
Multiple modalities retrieval	**EOPEN**	0.906	1.0	1.0	**0.969**
	Borda	0.906	1.0	1.0	**0.969**
	Reciprocal	1.0	0.649	1.0	0.883

Condorcet	1.0	0.947	0.947	0.965

3 DATA FUSION FOR EO CONTENT FOR INFORMATION RETRIEVAL

This section tackles the fusion and retrieval of multimodal EO content collected within EOPEN. As EO content, we consider the satellite products from the Copernicus Sentinel-2 mission that are annotated with multiple classes of the Corine Land Cover map. This section focuses on presenting the techniques that allow efficient retrieval of such content in response to a Sentinel-2 query. Each Sentinel-2 image is equipped with several modalities, including visual information (i.e. the RGB and other channels), and spatiotemporal information (i.e. the sentinel metadata that refer to the area that is depicted using geographical coordinates and timestamp that the image is taken) and the aim of this task is to consider all the aforementioned modalities, fuse them end eventually return similar content.

3.1 Related work

In remote sensing image retrieval task both traditionally extracted features and Convolutional Neural Networks have been investigated with the latter ones presenting performance advantage.

Specifically, CNN models that aim for both classification prediction and similarity estimation, called classification-similarity networks (CSNs), outputs class probability predictions and similarity scores at the same time (Liu et al., 2020). In order to further enhance performance, the authors combined information from two CSNs. “Double fusion” is used to indicate “feature fusion + score fusion”.

Moreover, Liu et al. (2017) proposed a feature-level fusion method for adaptively combining the information from lower layers and FC layers, in which the fusion coefficients are automatically learned from data, and not designed beforehand. The fusion is performed via a linear combination of feature vectors instead of feature concatenation.

Another work is that of Wang et al. (2016), who performed multiple SAR-oriented visual features extraction and estimated the initial relevance scores. For the feature extraction, they constructed two bag-of-visual-words (BOVWs) features for the SAR images and another SAR-oriented feature, the local gradient ratio pattern histogram. The authors calculated a set of initial relevance scores and constructed the modal-image matrix, then they estimated the fusion similarity and eventually reranked the results returned based on this similarity.

Finally, Li et al. (2016) used multiple type of features to represent high-resolution remote sensing images. One fully connected graph and one corresponding locally connected graph were constructed for each type of feature. Furthermore, a fused graph was produced by implementing a cross-diffusion operation on all of the constructed graphs. Then, from the fused graph, the authors obtained an affinity value between two nodes that directly reflects the affinity between two corresponding images. Eventually, in order to retrieve the similar images retrieval, the affinity values between the query image and the other images in the image dataset are calculated.

3.2 Methodology

For the retrieval of most relative content in EO data content, three different modalities where combined, each one representing a different aspect of the images of the dataset. Similarity by 'visual content', 'visual concepts' and 'geolocation and time' are explored. Eventually they are fused to all together to extract a more precise discrimination among the classes of the dataset. Each modality returns a ranked similarity list. Performing late fusion on the formed lists returns the final sorted list with the closest images to the given query-image.

Supervised methods applied for the training of a custom and some well-known pretrained deep neural networks using Keras library. A more classic approach to extract features based on colour histogram of various bands of an image was also tested. The MongoDB is holding the images metadata allowing the execution of geolocation and time queries.

3.2.1 Similarity by visual content

The aim of this module is to generate a model that transforms images into embedding vectors where the Euclidean distance between vectors represents how visually similar the images are as it regards to the content. To extract the necessary features from satellite patches two different approaches were evaluated. The first involved deep neural networks more specifically three pre-trained ImageNet DCNNs and a custom DNN. The second involved a more classical method that relies on feature extraction.

We selected 10 test images for each of the seven classes parsed from the Corine Land Cover inventory, and thus we ended up with 70 test image patches. Adetailed description of the dataset and the classes selected can be found in Section 3.3.1 The procedure followed for obtaining the similarity according to the visual content involves the following steps: a) we extract feature vectors for each patch of the dataset, including the test images, b) we calculate the distance between the query image and the rest images of the dataset, c)we retrieve the imageswith the lowest distance from the query-test patch, and d) we calculate the mAP for the top 30 results. . The full procedure is described in detail below.

For the feature extraction, we used layers closer to the top layers. Tested both with some well-known pre-trained networks and on a custom deep neural network:

A) Deep neural networks:
- **Pretrained networks**: Extracting features directly from specific intermediate layers of pretrained VGG19, ResNet-50 and Inception-ResNet-v2 networks. Three channel images were used as input. Since ImageNet is a dataset of RGB images we created an input dataset of same type of images. Red (band 4), Green (band 3) and Blue (band 2) Sentinel-2 bands are combined to form 3-channeled patches.
- **Custom Deep Neural Network**: Trained a DNN network (see layers below) with a structure that resembles VGG. It contains blocks of convolutional layers with 3x3 filters followed by a max pooling layer. This pattern is repeating with a doubling in the number of filters with each block added. The model will produce a 7-element vector with a prediction between 0 and 1 for each output class. Since it is a multi-label problem, the sigmoid activation function was used in the output layer with the binary cross entropy loss function. For input we tested with both 3 channel images (as done with the pretrained networks) and also with images that consisted of 5

bands of Sentinel 2 images, i.e. the Red (band 4), Green (band 3), Blue (band 2) for the 3-channel input, with the addition of NIR (band 8) and SWIR (band 11) for the 5-channel input. After the training we extracted features of the 5-channel patches from some intermediate layers.

In Table 2 the summary of the used DNN is depicted. For each layer the output shape and the number of the trained parameters can be observed.

Table 2: Layers summary of the Deep Neural Network

Layer type	Output Shape	Parameters #
Conv2D	(None, 120, 120, 30)	1380
Conv2D	(None, 120, 120, 30)	8130
MaxPooling2D	(None, 60, 60, 30)	0
Dropout	(None, 60, 60, 30)	0
Conv2D	(None, 60, 60, 60)	16260
Conv2D	(None, 60, 60, 60)	32460
MaxPooling2	(None, 30, 30, 60)	0
Dropout	(None, 30, 30, 60)	0
Conv2D	(None, 30, 30, 120)	64920
Conv2D	(None, 30, 30, 120)	129720
MaxPooling2	(None, 15, 15, 120)	0
Dropout	(None, 15, 15, 120)	0
Flatten	(None, 27000)	0
Dense	(None, 120)	3240120
Dropout	(None, 120)	0
Dense	(None, 7)	847

B) **Color histogram**: The histograms of a stack of bands were concatenated in a single vector. The same dataset of 3-channel and 5-channel images is used in here as well.

Three well known DCNNs were used for the feature extraction. For all of them we loaded a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as everyday objects and animals. As a result, the network has learned rich feature representations for a wide range of images.

VGG-19

VGG-19 is a convolutional neural network that is 19 layers deep. The network has an image input size of 224-by-224. Features extracted from fc1 (dense) and fc2 (dense) layers, with feature size of 1 x 4096 float numbers per patch. The architecture of VGG19 is depicted in Figure 14.

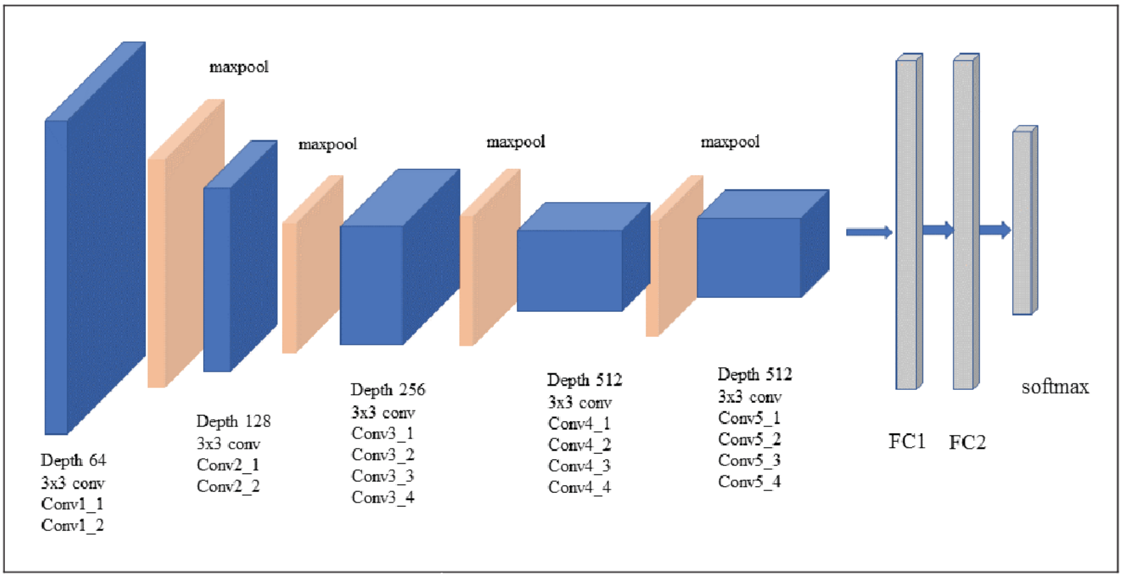

Figure 14: VGG19 architecture

ResNet-50

ResNet-50 is a convolutional neural network that is 50 layers deep. The network has an image input size of 224-by-224. Features extracted from avg_pool (GlobalAveragingPooling2) layer, with feature size of 1 x 2048 float numbers per patch.

Inception-ResNet-v2

Inception-ResNet-v2 is a convolutional neural network that is 164 layers deep and is formulated based on a combination of the Inception structure and the Residual connection. The network has an image input size of 299-by-299. Features extracted from avg_pool (GlobalAveragingPooling2) layer, with feature size of 1 x 1536 float numbers per patch.

Deep neural network

For the visual similarity we need information of the "inside" layers of the NNs. For each network we extracted features from one or more intermediate lower level layers of the model in order to get the vectors that best describe each image just before the final prediction layer.

To obtain the best possible results we evaluated with did hyper parameter optimization and enabled dropout regulation. We used the models with the best validation scores at a 5-Fold Cross-Validation, achieved with following settings (Table 3):

Table 3: Best scores and settings for 3ch and 5ch input of the DNN

Input Channels	Optimizer	Learning Rate	Batch Size	Epochs	F-Beta Score	F1 Score	Loss
3ch	Adam	0.001	128	200	0.843	0.843	0.232
5ch	Adam	0.0005	256	200	0.864	0.861	0.214

Colour Histogram

Apart from the DCNN-based features, we have investigated colour Histogram that was used in Candela Project[10]. In A.2 Appendix, there is a detailed table, which summarizes the aspects of the Similarity retrieval module in the EOPEN and Candela EU projects and thus makes clear the similarities and differences of the approaches proposed. In Candela project, raw Sentinel-2 band values were concatenated to form a long feature vector for each image query and normalisation was not applied. Thus, given that the data type of the initial bands is Uint16 meaning, the values may vary from 0-65,535. However, for the selected dataset the maximum value detected for any of the tiff files was below 21,000. These values were considered as the sub-vector length in order to reduce the concatenated vectors size, while at the same time preserving all the initial information. Thus, vector size for each query-image now is 1 x 63,000 for the 3 bands vectors (B04, B03, B02 Sentinel-2 bands) and 1 x 105,000 for the 5 bands vectors (B04, B03, B02, B08, B11 Sentinel-2 bands).

As a similarity measure the Euclidian Distance was used. For each query-image, its feature vector was generated and then the distance from all the feature vectors of the dataset was calculated. The *top-k* results with the less distance were kept.

3.2.2 Similarity by time and geolocation metadata

It is important to be able to find images that are timely and locally close to the query-image. For a quick retrieve of close images to the query we have used MongoDB queries. We have extracted all patches' datetime and geolocation metadata and inserted them in a MongoDB collection in the form of IsoDate and GeoJson respectively. Mongo allows quick indexing of the above data types. The Euclidean distance of the geolocations was used and then for images with the same distance, sorting by datetime was applied.

3.2.3 Similarity by visual concepts

In this module the concepts of an image are extracted. The methodology uses the deep neural networks of the Visual Content analysis module. This time we are using the last layer of each network that is responsible for predicting the class of the query-image and extract them as a vector. The Euclidean distance between vectors represents how visually similar the images are as it regards to the concept.

3.2.4 Fusion of modalities

In order to provide more consistent and accurate results on the retrieval task we combined the best performing methodology of Section 2.1.4 the previous sections, i.e. the similarity of visual content, visual concept and similarity by geolocation and time. For the fusion of the results we tested our algorithm against 3 know rank fusion algorithms; Borda count, Reciprocal and Condorcet fusion. For the extraction modality we used the VGG19 features, for the concept extraction the 5-channel custom DNN network's predictions. In our approach, that was described in detail in Section 2.1.4 , the fusion model was fed with the top 280 results of each of the previous modalities, due to limitations of MongoDB at the aggregation of the minimum distance of the geolocation query. For the evaluation of the various fusion methods we used the mean Average Precision (mAP) metric on the *top-30* results that were retrieved. Since our fusion method returns only strong candidates, there

[10] http://candela-h2020.eu/

are cases that the returned results are fewer than the top-30 limit. When this happens the *top-N* results that are returned by our fusion methodology query is the length that we use for the rest fusion methods for this specific query.

3.3 Results and Discussion

3.3.1 Quantitative analysis

Dataset Description

The BigEarthNet (http://bigearth.net/) dataset was selected for our experiments. The dataset contains ground-truth annotation about Sentinel 2 level-2A satellite images and consisted of 590,326 patches. Each image patch was annotated by the multiple land-cover classes (i.e., multi-labels) that were extracted from the CORINE Land Cover inventory of the year 2018 (CLC 2018).

Based on the available Corine land cover classes we did the following grouping of the closely related sub-classes of the CLC, forming seven major classes. We selected around 130,000 patches, of resolution 120 x 120 pixels in order to preserve a balance among the number of items of the different classes/concepts:

- Class rice: ‘Rice field’
- Class urban: ‘Continuous urban fabric’, ‘Discontinuous urban fabric’
- Class bare rock: ‘Bare rock’
- Class vineyards: ‘Vineyards’
- Class forest: ‘Broad-leaved forest’, ‘Mixed forest’, ‘Coniferous forest’
- Class water: ‘Water courses’, ‘Water bodies’, ‘Sea and ocean’
- Class snow: ‘Snow’

The selected classes are covering various PUCs subjects by including labels like water, rice and snow.

Results

The results for the feature extraction for the pretrained and the custom neural networks are shown at Table 4 using Mean Average Precision as metric and are computed against the Corine Land Cover (CLC) annotation of the BigEarthNet dataset. The VGG19 full convolutional layers are providing the best features for the retrieval problem. The flatten layer is underperforming. ResNet50 comes second with the Inception-ResNet_v2 falling behind. The custom DNN can’t catch up with the performance of the pretrained networks. But the significance of the extra bands is apparent when comparing 5-channel to 3-channel input.

Table 4: Mean average precision comparison on Feature extraction of seven classes among Pretrained networks, custom DNN and Colour Histogram methodologies.

	Pretrained Deep Neural Networks					Custom Deep Neural Network				Color Histogram	
classes	VGG19 fc2	VGG19 fc1	VGG19 flatten	ResNet50 avg_pool	Inception-ResNet_v2 avg_pool	5 bands flatten	5 bands dense	3 bands flatten	3 bands dense	3 bands	5 bands

top #10											
forest	83.02%	84.05%	81.17%	81.66%	63.70%	76.52%	80.38%	49.22%	50.89%	85.28%	85.83%
rice	86.79%	85.00%	75.28%	57.68%	29.21%	25.89%	17.41%	30.40%	11.57%	36.19%	51.48%
rock	62.21%	62.80%	76.38%	59.04%	58.09%	58.37%	52.96%	86.56%	60.44%	71.68%	70.12%
snow	86.37%	86.65%	43.96%	91.85%	88.46%	74.93%	87.79%	48.03%	79.57%	90.77%	90.50%
urban	68.22%	60.53%	45.46%	68.25%	73.43%	73.71%	66.53%	34.60%	42.77%	76.09%	70.95%
vine	74.74%	79.58%	76.07%	67.85%	42.75%	45.44%	47.78%	59.67%	39.51%	74.09%	87.58%
water	98.78%	100.00%	100.00%	100.00%	96.20%	100.00%	97.11%	95.22%	92.68%	93.81%	97.79%
mAP	**80.02%**	79.80%	71.19%	75.19%	64.55%	**64.98%**	64.28%	57.67%	53.92%	75.41%	**79.18%**
top #20											
forest	78.72%	82.37%	80.07%	77.63%	62.08%	76.12%	70.72%	45.98%	51.55%	82.75%	83.43%
rice	82.09%	81.77%	72.58%	49.74%	31.58%	21.01%	15.58%	30.40%	12.80%	32.32%	45.03%
rock	50.41%	53.52%	62.01%	51.59%	50.85%	46.30%	44.57%	83.94%	54.99%	62.80%	68.45%
snow	81.07%	80.39%	44.04%	90.92%	88.09%	74.52%	81.62%	49.20%	66.76%	90.00%	88.47%
urban	61.27%	53.65%	40.80%	64.92%	70.20%	69.26%	60.82%	30.85%	38.54%	74.52%	70.30%
vine	65.77%	69.36%	70.44%	61.53%	41.98%	41.55%	43.53%	44.75%	34.45%	67.64%	80.43%
water	98.83%	99.63%	100.00%	99.66%	97.00%	99.89%	96.58%	96.01%	92.27%	91.02%	95.98%
mAP	74.02%	**74.39%**	67.13%	70.86%	63.11%	**61.24%**	59.06%	54.45%	50.19%	71.58%	**76.01%**
top #30											
forest	76.29%	81.08%	78.35%	76.66%	62.18%	75.57%	68.98%	42.66%	49.57%	81.78%	82.06%
rice	78.30%	77.92%	70.95%	46.78%	25.43%	18.39%	14.51%	30.40%	12.09%	27.93%	39.37%
rock	44.89%	50.31%	55.56%	49.26%	47.75%	39.45%	42.23%	76.41%	45.12%	59.42%	65.86%
snow	78.21%	77.31%	41.58%	88.92%	87.23%	74.40%	80.00%	48.13%	61.59%	88.76%	86.18%
urban	55.35%	50.58%	38.94%	63.60%	67.74%	67.07%	58.06%	29.58%	36.36%	71.23%	69.29%
vine	60.61%	63.57%	63.52%	57.70%	42.12%	40.76%	42.03%	36.70%	33.94%	63.29%	76.19%
water	98.94%	99.48%	100.00%	99.22%	97.54%	99.72%	95.90%	96.14%	92.22%	89.25%	95.22%
mAP	70.37%	**71.47%**	64.13%	68.88%	61.43%	**59.34%**	57.39%	51.43%	47.27%	68.81%	**73.45%**

Similar conclusions when using accuracy at K as metric as shown in Table 5.

Table 5: Accuracy at K comparison on Feature extraction of seven classes among Pretrained networks, custom DNN and Colour Histogram methodologies.

Pretrained Deep Neural Networks						Custom Deep Neural Network				Color Histogram	
classes	VGG19 fc2	VGG19 fc1	VGG19 flatten	ResNet50 avg_pool	Inception-ResNet_v2 avg_pool	5 bands flatten	5 bands dense	3 bands flatten	3 bands dense	3 bands	5 bands
top #10											
forest	75.00%	77.00%	78.00%	73.00%	54.00%	71.00%	66.00%	38.00%	43.00%	80.00%	82.00%
rice	79.00%	81.00%	36.00%	38.00%	17.00%	12.00%	12.00%	13.00%	7.00%	15.00%	28.00%
rock	37.00%	40.00%	30.00%	43.00%	44.00%	23.00%	34.00%	20.00%	14.00%	53.00%	61.00%
snow	73.00%	72.00%	40.00%	86.00%	86.00%	73.00%	75.00%	38.00%	52.00%	89.00%	85.00%
urban	46.00%	43.00%	33.00%	59.00%	64.00%	66.00%	53.00%	22.00%	30.00%	70.00%	62.00%

vine	54.00%	57.00%	32.00%	53.00%	38.00%	27.00%	38.00%	19.00%	20.00%	60.00%	77.00%
water	99.00%	99.00%	100.00%	100.00%	97.00%	100.00%	97.00%	96.00%	92.00%	91.00%	95.00%
Average	**66.14%**	67.00%	49.86%	64.57%	57.14%	53.14%	**53.57%**	35.14%	36.86%	65.43%	**70.00%**
top #20											
forest	69.50%	78.50%	72.00%	73.50%	61.00%	73.50%	63.00%	33.50%	39.50%	78.00%	78.00%
rice	67.00%	68.00%	21.50%	30.00%	15.50%	10.50%	9.00%	6.50%	8.00%	10.50%	21.50%
rock	33.00%	36.00%	20.00%	42.00%	40.00%	18.00%	33.00%	10.50%	9.50%	48.00%	55.00%
snow	69.00%	69.00%	35.50%	82.50%	82.50%	74.00%	76.50%	35.50%	46.00%	86.00%	82.00%
urban	45.00%	41.00%	31.00%	59.00%	61.00%	61.50%	49.00%	23.50%	25.00%	64.00%	64.00%
vine	46.00%	50.50%	20.00%	47.00%	39.50%	26.50%	35.00%	13.50%	19.00%	53.00%	65.50%
water	99.00%	99.00%	100.00%	98.50%	98.00%	99.50%	94.50%	97.00%	91.50%	86.00%	93.50%
Average	61.21%	**63.14%**	42.86%	61.79%	56.79%	**51.93%**	51.43%	31.43%	34.07%	60.79%	**65.64%**
top #30											
forest	70.33%	76.33%	69.67%	72.00%	61.67%	73.33%	64.33%	36.00%	37.67%	78.67%	79.33%
rice	58.33%	60.33%	15.33%	24.67%	17.00%	11.00%	10.33%	4.33%	7.33%	10.00%	20.00%
rock	31.00%	33.33%	16.67%	40.33%	39.00%	17.00%	29.67%	8.00%	8.33%	48.00%	50.67%
snow	69.67%	68.00%	35.33%	80.67%	80.33%	74.00%	76.33%	33.33%	44.00%	84.00%	80.33%
urban	43.00%	41.33%	29.67%	57.67%	60.00%	58.33%	49.00%	20.33%	24.67%	62.67%	60.33%
vine	41.67%	46.67%	18.33%	48.33%	42.67%	26.00%	35.33%	12.00%	15.33%	48.33%	60.33%
water	99.00%	99.00%	100.00%	98.00%	98.67%	99.33%	94.67%	96.00%	92.33%	83.00%	93.00%
Average	59.00%	**60.71%**	40.71%	60.24%	57.05%	51.28%	**51.38%**	30.00%	32.81%	59.24%	**63.43%**

The results of the colour histogram are comparable with the best results obtained by the VGG layers.

Among the pretrained networks, the VGG19 fc2 layer managed to extract the best features for the task (Figure 15). The color histogram managed to outperform the VGG19 at top 20 and top 30 results, increasing the mAP score (Figure 16). On contrary the DNN modes with the 5-channel input was unable to follow (Figure 17).

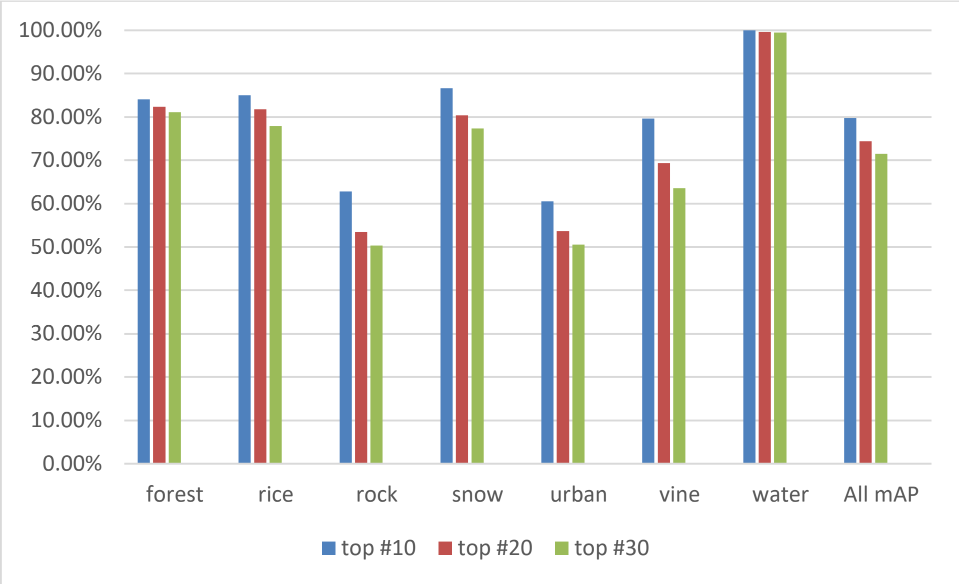

Figure 15: Results at mAP metric on Feature extraction using VGG19 fc1 layer.

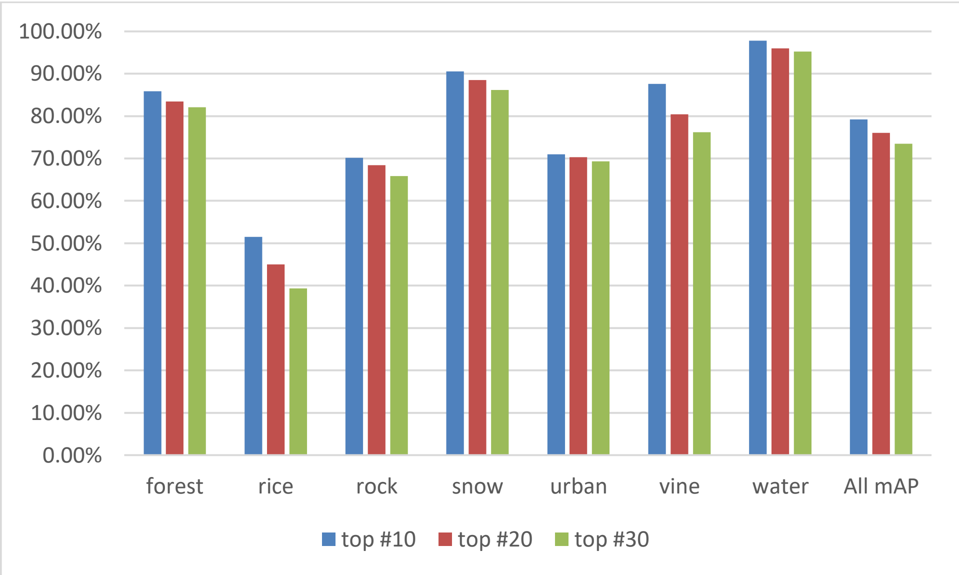

Figure 16: Results at mAP metric on Feature extraction using Color Histogram.

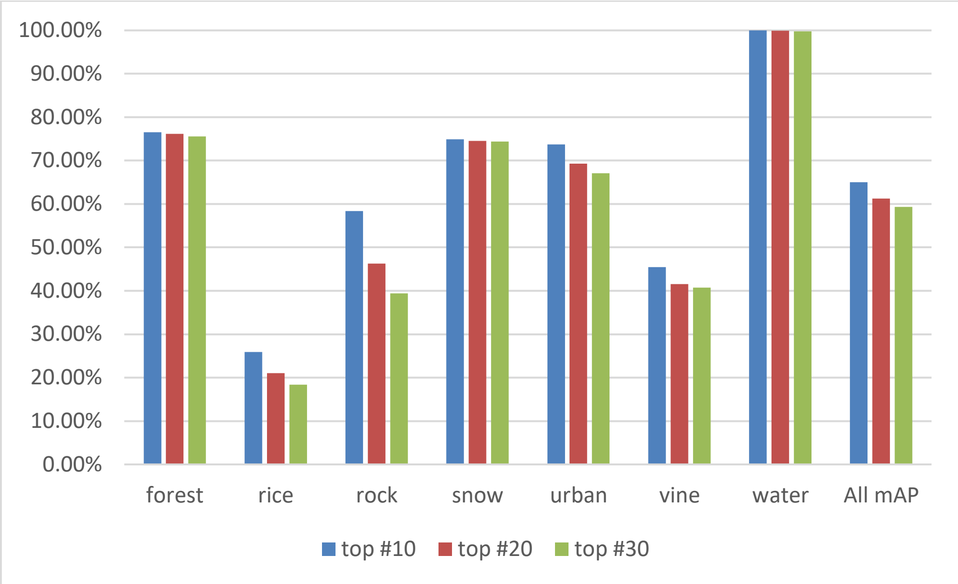

Figure 17: Results at mAP metric on Feature extraction using DNN 5 channel.

Comparing the pretrained VGG19 with the Colour Histogram methodology on most classes they perform the same. The only point that they greatly differentiate is that at rice identification VGG19 performs much better, while the color histogram method performs better at the Urban and at the rock areas.

Concept extraction

The results for the concept extraction for the pretrained and the custom neural networks are shown at Table 6 using Mean Average Precision as metric. Here the concepts are extracted directly by the last prediction layer. The 5-channel custom DNN presents the best results. The 3 channel one seems to be missing crucial information falling behind. For the pretrained networks the Inception-ResNet_v2 outperforms ResNet50 and VGG-19.

Table 6: Mean average precision comparison on Concept extraction of seven classes among Pretrained networks, custom DNN and Colour Histogram methodologies.

Pretrained Deep Neural Networks				Custom Deep Neural Network		
classes	VGG19 predictions	ResNet50 fc1000	Inception-ResNet_v2 fc1000	classes	5 bands dense (last)	3 bands dense (last)
top #10						
forest	63.59%	71.28%	66.67%	forest	80.38%	49.22%
rice	19.60%	3.25%	34.16%	rice	17.41%	30.40%
rock	22.70%	14.13%	30.68%	rock	52.96%	86.56%
snow	63.48%	84.47%	91.14%	snow	87.79%	48.03%
urban	55.66%	69.17%	58.85%	urban	66.53%	34.60%
vine	44.78%	29.43%	48.55%	vine	47.78%	59.67%
water	93.44%	97.47%	99.77%	water	97.11%	95.22%

mAP	51.89%	52.74%	**61.40%**	All mAP:	**64.28%**	57.67%
top #20						
forest	61.38%	66.58%	60.16%	forest	70.72%	45.98%
rice	18.49%	3.54%	27.07%	rice	15.58%	30.40%
rock	22.98%	15.85%	28.73%	rock	44.57%	83.94%
snow	62.33%	83.91%	87.47%	snow	81.62%	49.20%
urban	54.03%	65.62%	54.58%	urban	60.82%	30.85%
vine	39.65%	28.15%	43.32%	vine	43.53%	44.75%
water	93.49%	96.73%	98.52%	water	96.58%	96.01%
mAP:	50.33%	51.48%	**57.12%**	All mAP:	**59.06%**	54.45%
top #30						
forest	59.34%	64.78%	58.30%	forest	68.98%	42.66%
rice	16.92%	4.75%	27.02%	rice	14.51%	30.40%
rock	22.13%	16.18%	26.54%	rock	42.23%	76.41%
snow	62.42%	83.75%	86.36%	snow	80.00%	48.13%
urban	52.17%	63.43%	48.42%	urban	58.06%	29.58%
vine	35.01%	26.82%	39.82%	vine	42.03%	36.70%
water	93.76%	95.95%	98.32%	water	95.90%	96.14%
mAP:	48.82%	50.81%	**54.97%**	All mAP:	**57.39%**	51.43%

Conclusions vary when using accuracy at K as metric (

Table 7). Here, the best results are demonstrated by Inception-ResNet_v2 and ResNet50, followed by 5-channel custom DNN.

Table 7: Accuracy at K comparison on Concept extraction of seven classes among Pretrained networks, custom DNN and Colour Histogram methodologies.

	Pretrained Deep Neural Networks			Custom Deep Neural Network	
classes	VGG19 predictions	ResNet50 fc1000	Inception-ResNet_v2 fc1000	5 bands dense (last)	3 bands dense (last)
top #10					
forest	57.00%	62.00%	54.00%	61.00%	24.00%
rice	10.00%	2.00%	13.00%	2.00%	2.00%
rock	17.00%	13.00%	17.00%	30.00%	3.00%
snow	57.00%	84.00%	81.00%	77.00%	17.00%
urban	39.00%	60.00%	40.00%	34.00%	24.00%
vine	23.00%	20.00%	33.00%	10.00%	4.00%
water	94.00%	96.00%	97.00%	89.00%	87.00%
Average	42.43%	**48.14%**	47.86%	**43.29%**	23.00%
top #20					

forest	54.50%	58.00%	53.50%	62.50%	22.00%
rice	9.00%	2.00%	12.50%	2.50%	2.50%
rock	17.00%	14.50%	15.00%	33.50%	3.50%
snow	61.00%	83.50%	79.50%	76.50%	19.00%
urban	36.00%	59.50%	39.00%	34.50%	25.50%
vine	20.00%	21.50%	29.50%	13.50%	8.50%
water	94.00%	94.50%	97.50%	87.00%	84.50%
Average	41.64%	**47.64%**	46.64%	**44.29%**	23.64%
top #30					
forest	53.67%	60.00%	52.33%	59.00%	22.33%
rice	10.67%	3.67%	9.67%	2.33%	2.33%
rock	16.67%	13.33%	13.67%	31.67%	3.33%
snow	62.67%	83.33%	75.33%	75.00%	20.00%
urban	35.67%	56.33%	39.33%	34.00%	26.00%
vine	20.33%	22.67%	29.00%	17.33%	10.33%
water	94.00%	93.67%	98.33%	86.33%	83.33%
Average	41.95%	**47.57%**	45.38%	**43.67%**	23.95%

Among the pretrained networks, the Concepts Inception-ResNet v2 (Figure 18) managed to extract the best concepts for the task. The DNN model with the 5-channel input provided similar results (Figure 19Figure 17).

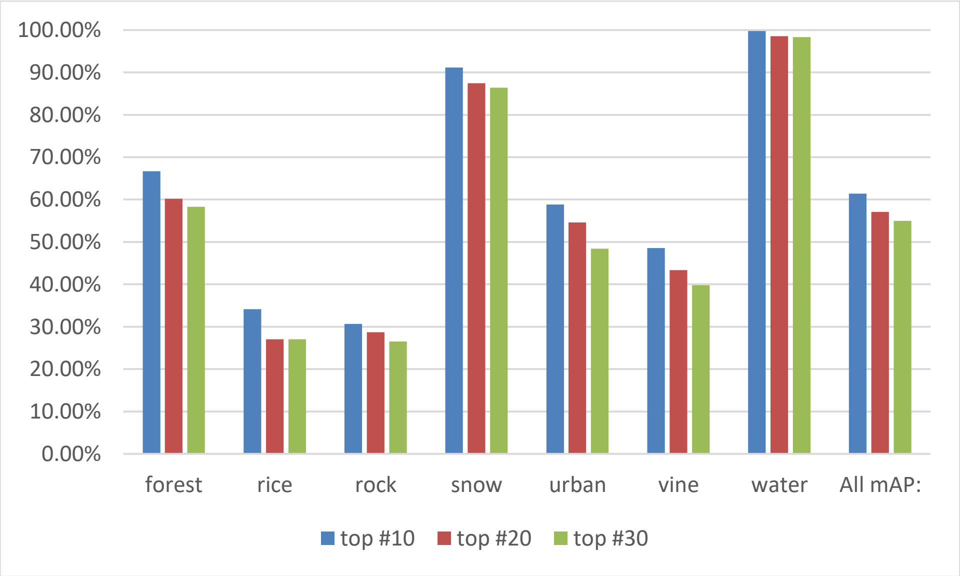

Figure 18: Concepts Inception-ResNet v2 - mAP

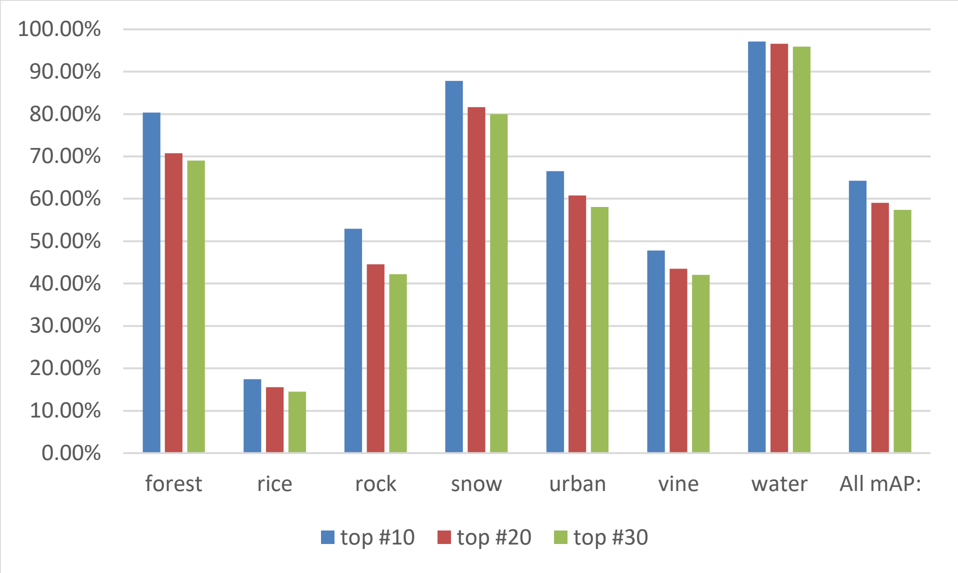

Figure 19: Concepts DNN-5ch - mAP

Fusion

Our fusion method performs marginally better than the Borda fusion, leading the metrics array in the comparison results (

Table 8). Reciprocal and Condorcet were not able to complete the other two methods in any class metric.

Table 8: Comparison of fusion methods with mean Average Precision metric

Classes \ Method	Ours	Borda	Reciprocal	Condorcet
forest	89.56%	88.38%	60.11%	52.85%
rice	97.05%	98.92%	39.51%	66.61%
rock	62.90%	64.69%	26.53%	20.88%
snow	91.46%	89.44%	67.04%	15.22%
urban	79.96%	74.90%	53.72%	29.03%
vine	88.40%	88.25%	28.22%	19.54%
water	97.35%	97.97%	78.42%	76.05%
mAP:	**86.67%**	86.08%	50.51%	40.03%

3.3.2 Qualitative analysis

Here we present for each of the seven classes one representative query and its top-10 similar images. The first patch is the image-query, whereas the following 10 images are the more similar patches. Each row represents a single class. In the following, we provide example results for all four fusion algorithms (Figure 20, Figure 21, Figure 22, Figure 23).

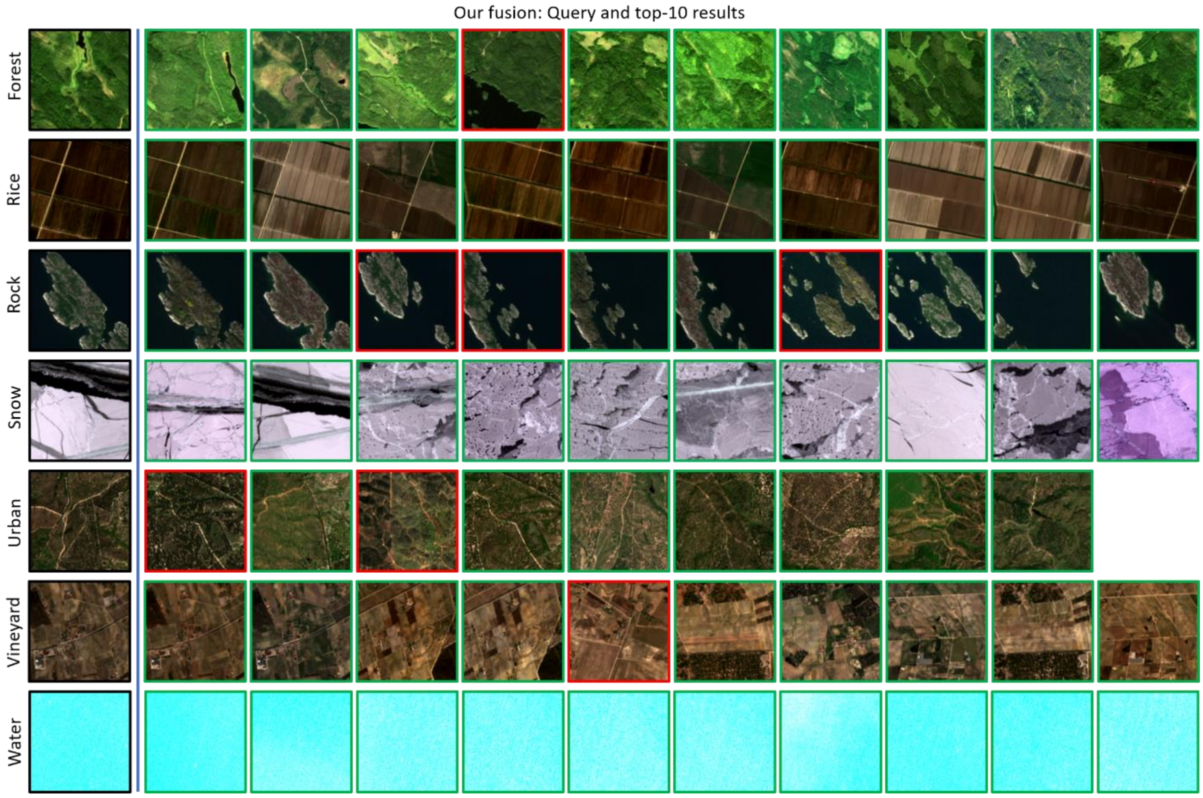

Figure 20: EOPEN fusion – Query and top - 10 results.

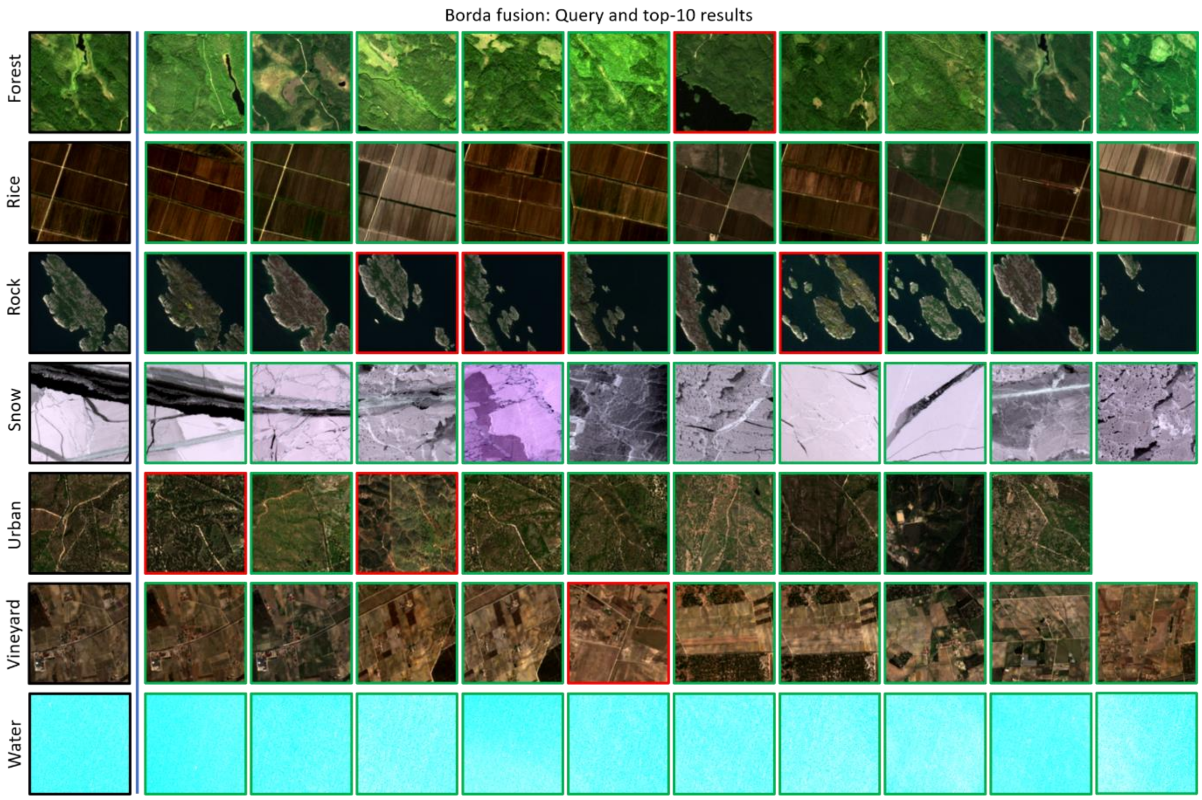

Figure 21: Borda fusion – Query and top - 10 results.

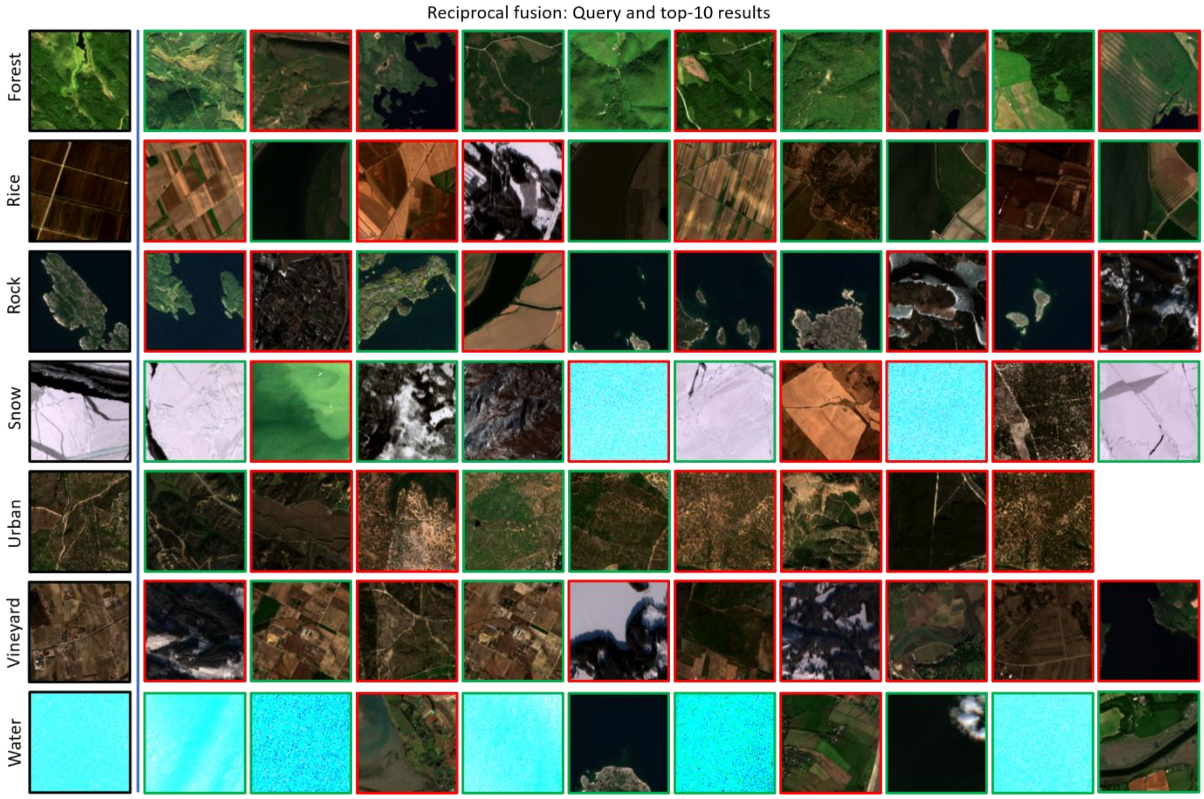

Figure 22: Reciprocal fusion – Query and top - 10 results.

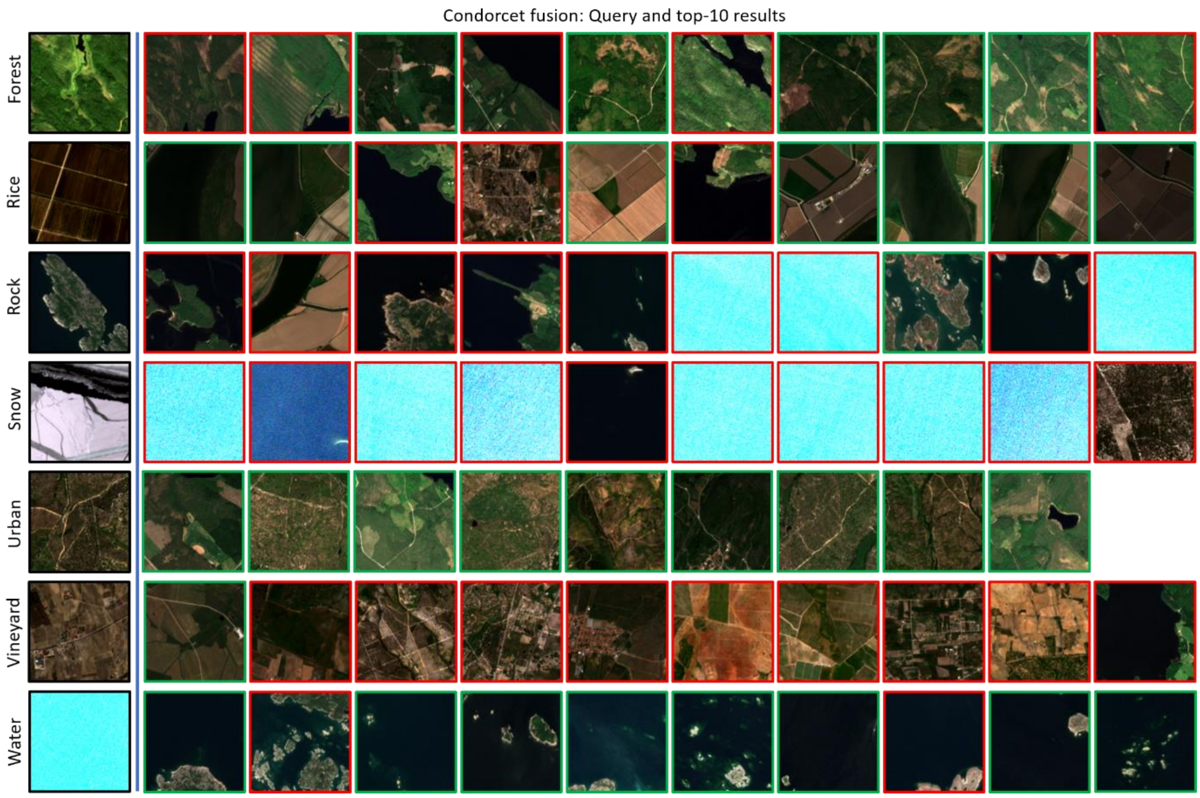

Figure 23: Condorcet fusion – Query and top - 10 results.

Based on the qualitative evaluation, the conclusions that can be drawn regarding the images misclassified in our approach.

Urban query: Most of the misclassified results are rice. Visually the two classes resemble to each other making difficult for the DCNNs to discriminate among them.

Forest query: The returned misclassified images are mostly water patches. Many queries contain water bodies like rivers or lakes.

Snow query: The returned misclassified images are mostly water. The main issue here is that some search-queries contain a small lake or river, resulting to the increased retrieval of water patches. Also, some of the retrieved images are of the forest class, because in some cases they depict sparse country-side areas mixed with snow.

Vine query: Almost all the misclassified images were actually urban patches. Visually, there is great similarity between these two classes.

Rock query: Rock queries are mostly rocky areas near water, resulting to fetching many water patches.

The overall conclusions that can be drawn from the experiments presented in this section are the following:

- For feature extraction VGG19 outperforms all the pretrained and custom DNNs that fail to catch the features with enough detail.
- For concept extraction, custom DNNs outperforms the other methods, while it comes second when comes to accuracy at K. Also, inception-ResNet_v2 and ResNet50 outperforms the VGG in this domain. In all cases the significance of moving from 5-channel to 3 channels is evident for the custom DNN as expected.
- For the fusion our method gives slightly better results than Borda fusion, with Reciprocal and Condorcet been proved inefficient for this task. Some classes are hard to be recognized cause many patches are multi-labelled with mixed characteristics or are difficult to discriminate due to great visual similarity.

4 FUSION OF SENTINEL AND SOCIAL DATA FOR SNOW DEPTH ESTIMATION

Northern European countries, such as Finland, experience a very long winter season which can last several months depending on the region, putting a lot of stress on infrastructure. Also, extreme snow events, besides the benefit it has on winter tourism, can also cause severe problems in electricity supply, traffic, and agriculture. Addressing these problems requires a well-organized civil protection agency, as well as significant financial resources.

Recently, civil protection agencies have adopted the use of remote sensing from a number of platforms (e.g. satellite images and UAVs) in extreme event management, with potential benefits in financial cost and decision making. However, very often the remote sensing data collected by these platforms are not available fast enough for the decision making required in such events. Additionally, malfunction of these platforms, or potentially deliberate attacks to ground segments of space systems, can lead to delayed decision making and ultimately can result in loss of life. Therefore, alternative data sources must be used to fill this gap. An example is given in Figure 24, with publicly available information on Twitter about snow observations.

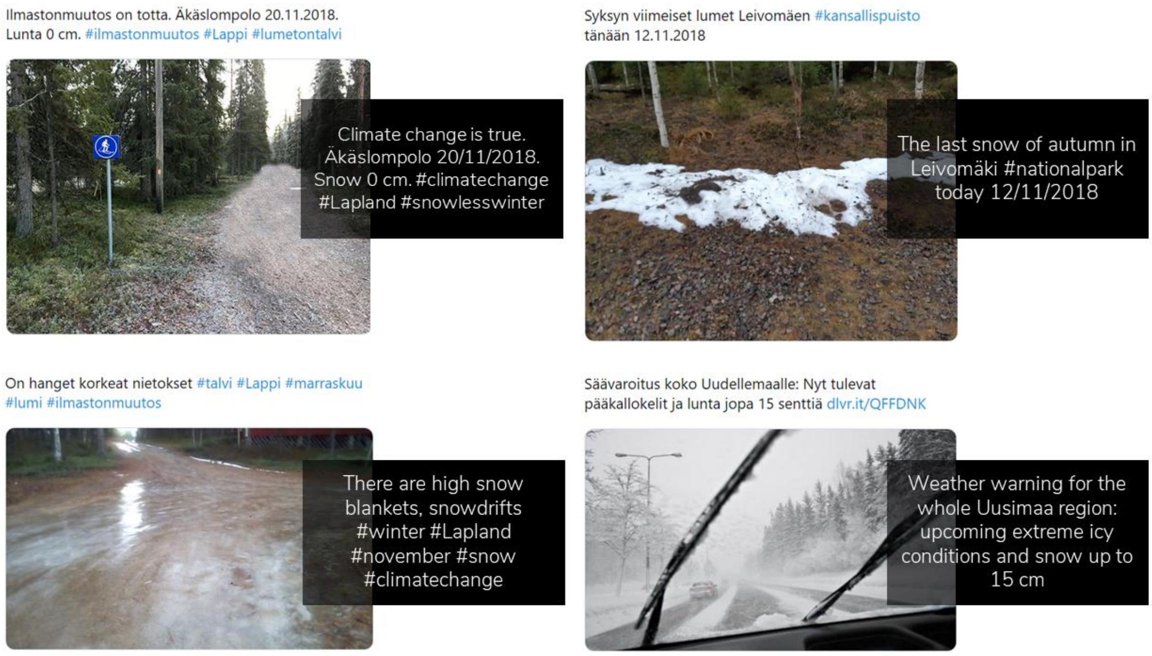

Figure 24: Example of online social media data from Twitter posts in Finland.

During the last decade, publicly available information from social media (e.g. Twitter, Facebook, Instagram, and various blogs) have made their debut in civil protection agencies around the globe. They can be used as a snap-shot of the public response to an extreme event and they are instant, with a huge benefit to decision making. It turns out that when it comes to breaking news, Twitter appears to have outperformed traditional media. During the Sichuan 2008 earthquake, which was responsible for 70.000 deaths, information regarding an initial tremor was being disseminated by Twitter several minutes before the main burst (Li et al.2008), and a similar case took place during the 2008 Southern California earthquake as well.

Twitter was also used to disseminate information immediately after the Haiti 2010 earthquake providing an insight regarding injured and trapped victims as well as damaged buildings (Oh, et al.2010). Pollution related health issues is another potential field where Twitter could be used given that the public response through this platform is strongly correlated with the ambient air quality in major urban areas (Gurajala et al.2019). Terrorist attacks are also a field where microblogging (e.g like Twitter) has been shown to play a vital role in collective sense-making immediately after the first shock, significantly contributing to awareness and reducing the uncertainty associated with such events (Haverin et al.2012). Twitter can also be an effective source of data in other extreme events scenarios. For example, it has been shown as an effective source of data that can be used to identify flood "hot-spots" immediately after these take place, which will, in turn, be used to task remote sensing data collection (Satellites and UAVs) for a more detailed analysis during the crisis management (Cervone, et al.2015).

Furthermore, recent event-detection efforts regarding floods have shown that today's algorithms can detect floods, including those caused by major storms or hurricanes, with great accuracy on a global level (Bruijn, et al. 2019). Wildfire is another natural disaster that has attracted the attention of civil protection agencies. Wang et al. (2016) has shown that monitoring social media could benefit such crisis management especially by increasing situational awareness, and helping with the evacuation, damage assessment and rescuing efforts.

Given that microblogging has evolved into such a widespread tool during a crisis, we attempt to expand its use and highlight its potential in managing snowfall events. Our goal is to highlight possible correlation between snow depth and the amount of microblogging associated with this event. This would suggest that the amount of the related information that is shared among the public could be used as a proxy for the intensity of the meteorological event. For this, besides using snow depth derived from ground measurements and model simulations (described in the next section), we also adopt a method of estimating the snow depth based on satellite images of backscatter radiation. The proposed methodology is validated in Northern Europe (Finland), but can be directly extended to other high-latitude areas of interest. Given that snow related tweets are just a small subset of a larger pool or tweets covering all types of social life, an important tool used in this study is artificial intelligence and machine learning utilized to annotate thousands of tweets, which are then fused with the geophysical data to enhance our snow depth estimate.

4.1 Related work

The estimation of snow depth has been very popular using Synthetic Aperture Radar (SAR) images collected by the Copernicus Sentinel-1 constellation (satellites 1A and 1B), due to the lack of in-situ data in the area of interest. Sentinel 1 is a SAR mission that provides 5m×20m resolution backscatter measurements in co-polarization and cross-polarization. These are ground range detected (GRD) Interferometric Wide Swath (IW) backscatter (at C-band; 5.4 GHz). Both Sentinel 1 satellites (1A and 1B) have the same orbital plane, but have a 6-day offset with each other. Each satellite has a 12-day repeat cycle, and 175 orbits per cycle. Because of this, each observation from the Sentinel 1 constellation has a different incident angle relative to a flat surface (ranging between 29 and 46).

Traditionally, SAR C-band backscatter satellite measurements have been used to study snow melt (Nagler et al., 2016; Nagler et al., 2018), based on the high dielectric loses of water that lead to a reduced backscatter coefficient over wet snow compared to surfaces that are snow-free or covered by dry-snow. For snow depth, C-band satellite backscatter measurements were used early on in the past, but only in co-polarization σ_{vv}^0, and showing only limited sensitivity (Bernier et al., 1999; Shi and Dozier, 2000).

On the other hand, cross-polarization backscatter σ_{vh}^0 has been used to estimate snow depth in the past, but only locally using tower installations (Kendra et al., 1998; Strozzi and Matzler, 1998). Recently, a new method that utilizes the ratio of co-polarization to cross-polarization backscatter $\sigma_{vh}^0/\sigma_{vv}^0$ has been implemented on Northern Hemisphere mountainous regions, exhibiting a promising snow depth estimate (Lievens et al., 2019). C-band σ_{vv}^0 measurements shows little variation during winter due to the limited absorption of scattering by dry snow, but exhibit a sharp decline during the melting period, due to the large absorption of backscatter by wet snow. In contrast, σ_{vh}^0 increases during winter as the snowpack intensifies. This is due to a raising path length of the radar signal, which results in increased backscattering. The logic behind the use of the ratio lies on the fact that it eliminates the effects of temporal changes in the ground surface, vegetation, and snow conditions, which affect both σ_{vv}^0 and σ_{vh}^0 the same way (Lievens et al., 2019). Furthermore, during winter snow accumulation (spring snow melt) increases (decreases) due to the higher increase (decrease) of σ_{vh}^0 compared to that of σ_{vv}^0.

Despite the fact that the cross-polarization ratio method was implemented on a larger scale for conditions of deep snow (up to 1-3 meters) and mostly bare ground at high elevation, we attempt to use this method for estimating the snow depth on a much smaller scale (30Km × 40Km) for the low land area around the city of Helsinki, where additional data from citizen observations may be fused to further enhance the estimation capacity of the new model. Contrary to existing approaches, we fuse Twitter data with the snow depth estimation model of Lievens et al. (2019) to further improve the estimation of snow depth in high-latitude areas. The developed methodology is presented in the following section.

4.2 Methodology

The proposed methodology combines Sentinel 1 images and Twitter data that are highly correlated with actual snow conditions. The method fully exploits the citizen interactions in their social system, as they are expressed by short text and images on social media platforms (e.g. Twitter). The overall framework is presented in Figure 25.

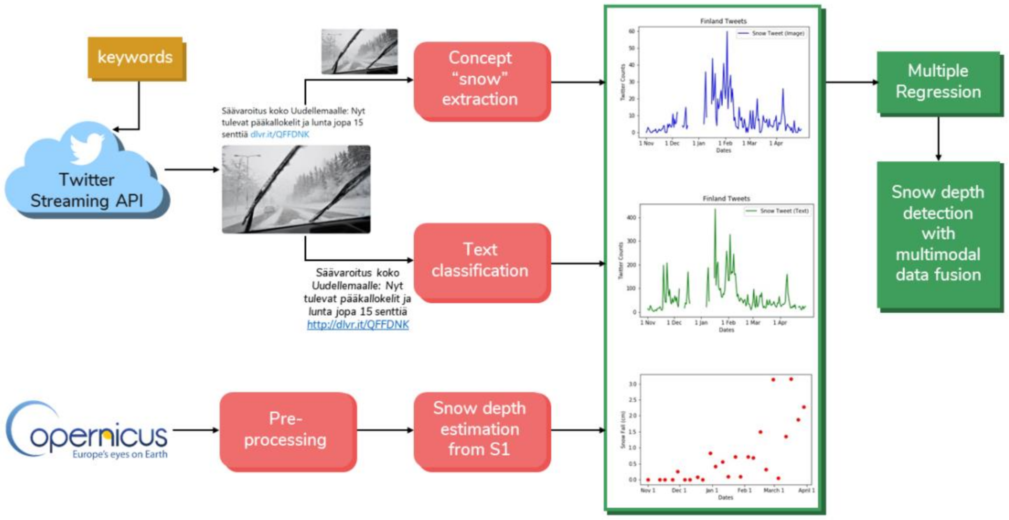

Figure 25: The proposed framework for snow depth estimation using EO and non-EO data.

4.2.1 Snow depth estimation using Earth Observation Data

The snow depth is estimated using Sentinel 1 images in the north of Helsinki. Preprocessing of the raw satellite data (Sentinel 1) is carried out through the Sentinel Application Platform (SNAP1) and includes radiometric calibration, speckle noise removal, terrain correction and linear to dB transform. Our area of interest is chosen more inland (north and northeast of Helsinki) to avoid the city centre, where snow is frequently cleared of the streets and roads. Two separate locations are chosen for this in order to validate the results from the satellite derived snow depth. Given that Southern Finland is covered to a large extent by forests, which attenuate backscatter, we mask out these areas till we are left with the open grassy and farmland areas. Additionally, we mask out water areas (lakes) regardless of whether the water bodies are frozen or not. Both forest and water have been masked out.

In order to properly process the backscatter data that often come from different orbits we need to remove the static bias. First, we average 1-year of backscatter data for each orbit separately, and then average for all orbits. The static bias for each orbit is estimated as the difference between the all orbit mean and the specific orbit mean, which is added to the backscatter time series for the corresponding orbit. Also, before computing their ratio, co-polarized and cross-polarized backscatter are re-sampled and projected onto a coarser 1Km^2 grid by linear averaging, which also helps reduce speckle noise. If more than 20% of the pixels, corresponding to the native Sentinel 1 resolution, are assigned as water or forest when they are projected onto the coarser grid, the 1Km^2 average is automatically removed from the analysis.

The first step in estimating the snow depth is to estimate the snow index as follows:

$$SI(i,t) = \begin{cases} max(0,[SI(i,t-1) + BR(i,t) - BR(i,t-1)]) & if\ SC(i,t) = 0 \\ 0 & if\ SC(i,t) = 0 \end{cases} \quad (1)$$

where SC is the snow cover, BR represents the ratio of backscatter radiation in cross-polarization (VH) and co-polarization (VV) if the backscatter is given in linear scale. However, if the backscatter is given in dB the difference must be used instead. Next, we rescale the snow index into snow depth with:

$$SD(i,t) = \left(\frac{a}{1 - bFC(i)}\right) SI(i,t) \tag{2}$$

where a = 1.1dB^{-1}, b = 0.6 and FC is the evergreen forest cover fraction (dimensionless).

4.2.2 Citizens observations about snow and reporting on social media

In this work we combine social media data from Twitter, that represent a public response to snow fall events and the accumulation of snow in the area of Finland, with the snow depth estimation, as it is presented in Equations (1) and (2).

Data Collection

11,024 tweets were collected, covering a period of 151 days, i.e. from November 2018 till March 2019. The Twitter Streaming API is used to collect relevant Twitter posts. The collection is keyword-based, where Finnish words for snow (e.g., lumi, lunta, lumeen) have been used as queries. Out of the collected data, 3,210 tweets have been manually annotated by the Finnish Meteorological Institute as relevant or irrelevant, based on whether the text content was indeed about snow weather or just included one of the keywords (e.g., a metaphor). This annotated dataset has been further used to train an algorithm that is able to automatically classify the text of a tweet as relevant or not.

Representation of Twitter text as a feature vector

Starting with the text representation, we used the state-of-art algorithm Bidirectional Encoder Representation from Transformers (BERT, Devlin et al. 2018). BERT involves an attention mechanism that learns contextual relations between words in a text. BERT's goal is to generate a language model, and the used mechanism reads the entire sequence of words at once, contrary to directional models (e.g. n- gram LMs (Rosenfeld, 2020), and neural network LMs (Mikolov et al. 2010; Bengio et al. 2003)) that read the text input sequentially. Therefore, it is considered bidirectional or non-directional. This characteristic allows the model to learn the context of a word based on its surroundings. In order to capture the text representation of the whole tweet, we used an existing pre-trained model in Finnish language called 'bert-base-finnish-cased-v1'. Thus, the input in the BERT model is a Twitter text and the output is a feature vector with a length of 768.

Logistic regression classification in Twitter text (tweet)

A Logistic Regression (LR) model is trained to classify Twitter posts (short text) as relevant or not to snow. This disambiguation allows the removal of tweets that refer to metaphoric meanings of the word "snow" or synonyms of it. The model is trained by using the manually annotated data provided by the Finnish Meteorological Institute and a grid search is realized in order to identify the best parameters. We report that a basic logistic regression model is developed with parameter C equal to 31.57. Eventually, the model is validated on a set of 11,024 new non-annotated data, and the model estimated 6,097 tweets (55%) as relevant and 4,927 (45%) as irrelevant.

Deep Learning for visual concept (snow) extraction on Twitter images

A frame- work different to text classification of the tweets has been used to extract visual concepts from the subset of tweets that include images. The target is to count the number of Twitter images per day that illustrate snow. To that end, we used a 22-layer GoogleNet network (Szegedy et al., 2015) that was trained on 5,055 ImageNet concepts (Pittaras et al., 2017). At this stage, the classification layer of the network, which is a fully connected layer, has dimension equal to 5,055, i.e. the total number of given concepts. Thus, this framework receives as input an image, then the fined-tuned Deep Convolutional Neural Network is tested on the specific image and a list of concepts along with their probabilities is produced. If the concept “snow” is ranked among the top-10 concepts with probability higher than 0.01, then we consider that the image contains the concept. Eventually, concepts are extracted from all the Twitter images, and 1,118 have been found to contain the concept “snow”.

4.2.3 Correlating social media observations with snow depth

The snow depth estimate can be, in some cases, strongly correlated with the actual measurements and weakly correlated in other. Even in the case when the correlation is significant, severe biases may exist due to an overestimation or underestimation of the actual measurements. To reduce these biases we use regression analysis:

$$\hat{Y} = \sum \alpha_i X_i + \beta_o \tag{3}$$

where $\hat{Y}$ is the predicted snow depth from a linear regression model, α_i and β_i are coefficients derived through regression analysis between the predictor variables X_i and the observed snow depth Y. In our proposed approach, X_1 is the estimated snow depth SD. By Twitter data X_1 we mean either the number of relevant tweets T_t per day t or the number of Twitter images I_t per day t that contain snow. Our proposed model is formulated as follows:

$$\hat{Y} = \alpha_1 SD(t) + \alpha_2 I_t + \alpha_3 T_t + \beta \tag{4}$$

The proposed model of Eq. (3) is using $SD(t)$, T_t and I_t, showing the added value of social-generated data assets, which is either estimated from Logistic regression classification in Twitter text or a Deep Convolutional Neural Network on Twitter images to count the number of posts that are relevant to snow. In the following Section, we compare the 1st order model which uses only satellite image-based estimations $SD(t)$, with the proposed 3rd order model that fuses state-of-the-art snow depth estimations, social media images and social media short text (tweets).

4.3 Results and discussion

Validation data in Finland

Regarding snow depth, we use observations, as well as simulations, for validation purposes. First, we use direct snow depth measurements from four sites around the city of Helsinki (Figure 26). These were taken using instruments (SR50AH) that measure snow depth by emitting an ultrasonic pulse and then measuring the elapsed time between the emission and return of the pulse. The instrument also uses air temperature measurements to correct for sound speed variations. Snow depth is measured every ten minutes, and then it is averaged to provide daily mean snow depth. All four sites are located in areas with intense urban development, which is ideal for this study given that we need to accurately represent the snowfall conditions of the area where the public response (Twitter posts) takes place.

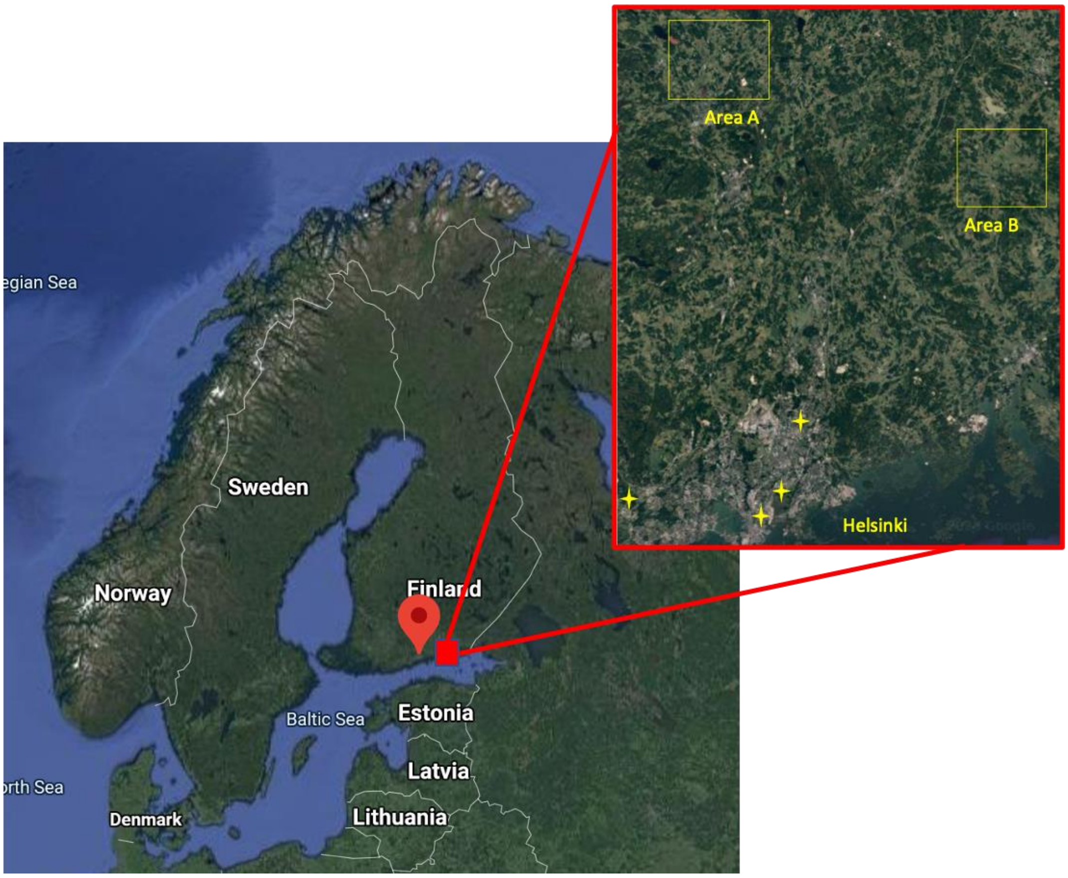

Figure 26: Area of interest and measurements from four pilot sites in Finland.

According to the snow depth measurements snow fall begins in mid-December and we have a constant snow build-up through the beginning of February due to several significant snow events (>4cm). This is followed by a melting period that extends till late April, with only one major snow event in between. Only one meteorological station (site 2) located in the centre of the city and close to the sea has a shorter (by roughly 10 days) snow cover period. All sites exhibit the same dynamic (variability), but differ in the snow amount, with the ones more inland experiencing heavier snowfall than the ones closer to the sea. The observed snow depth in the four considered sites and their average are illustrated in Figure 27.

Correlation between social media and snow depth.

For correlation purposes in the following analysis we use the average of all sites. A close inspection of the observed snow depth and the Twitter time series (text-based or image-based) we can see that the three largest snow events (during 09, 17, and 29 January 2019) coincide with peaks in tweeted snow images and number of relevant tweets which took place the same or the next day.

So far we have defined the variables T_t and I_t, which denote the number of relevant-to-snow tweets at day t and the number of Twitter images that contain snow at day t, respectively. We also denote by $\rho(A, B)$ the correlation coefficient between two variables A and B. The corresponding estimated Pearson correlation is denoted by $r(A, B)$, which results the following outcomes with the average observed snow depth Y:

• $r(Y, I_t)$= 0.56, positive and statistically significant correlation

• $r(Y, T_t)$ = 0.51, positive and statistically significant correlation

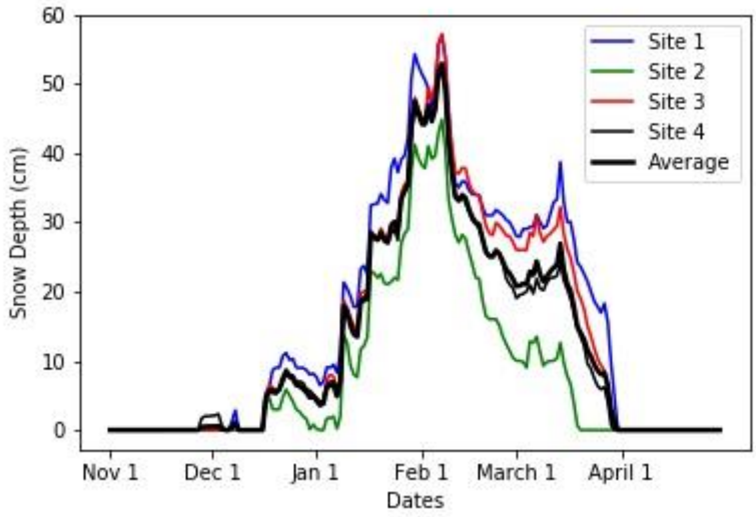

Figure 27: Daily snow depth from the four sites and their average during November 2018-March 2019.

Data fusion between social media and Sentinel 1 images.

The satellite derived snow depth of Equations (1) and (2), as expected, is less good and doesn't have the same time density and is only estimated based on Sentinel 1A measurements during Dec 2018-March 2019. During this period, Sentinel 1B satellite did not give any measurements for the area of Helsinki in IW mode, and are given in EW mode instead due to the Baltic Sea Ice campaign, limiting our snow depth estimate to every 6 days only in area A and a with slightly increased frequency in area B. Additionally, we exclude the snow depth computation during the period between late February to March, given the fact that the liquid water from melting snow during the melting period overwhelms any backscatter signal from shallow snow. The resulting estimated snow depth is poorly correlated with the observed one, i.e. r=0.25 for area A and r=0.41 for area B. This is most likely due to the fact that the snow was shallow in the area of Helsinki, which did not allow the Lievens' method to exhibit its full potential. However, here is exactly where the use of citizen observations through Twitter are fused with the satellite-based estimation to provide an improved estimation. The improvement is measured with the Mean Squared Error (MSE) evaluation measure that is able to quantify the deviation between the observed snow depth values with each model.

The positive correlation between the tweets and the snow observations is a necessary condition that allows us to fuse these two through a regression model in an attempt to improve our Sentinel 1 snow depth estimate. This can be seen in more detail as we build the regression model, based on the correlation between the observed snow depth and the number of tweets (variables I_t and T_t). The time series are shown in Figure 28.

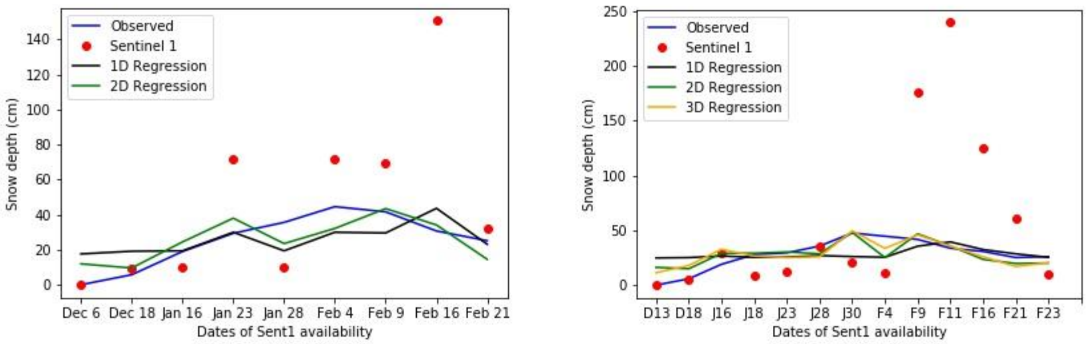

Figure 28: Variation of observed (ground truth) and satellite derived snow depth for Area A (left) and Area B (right). The estimated snow depth based on the 1D, 2D and 3D regression model is also shown. The data are only presented for the dates when the Sentinel 1 backscatter is available.

The Sentinel 1 derived snow depth exhibits a large mean square error (MSE) when compared to the actual snow depth. This is mainly the result of outlier points like the one during February the 16th (Figure 29). Such large deviations are mainly the result of backscatter noise that overwhelms the signal from the snow cover, which is relatively low in our case. A 1st order (1D) regression model, which uses only the estimated snow depth, can remove the effect of such outliers by utilizing the linear relationship between the predictor X_1 and the observed snow depth Y. The resulting snow depth $\hat{Y}$ has a much smaller MSE with no outliers. However, what needs to be examined is whether this can be improved by adding the information given by social media. It turns out that a 2nd order model (2D) expressed by the equation:

$$\hat{Y}_t = 0.17SD(t) + 0.98I_t + 6.07 \tag{5}$$

exhibits a smaller MSE, reduced by 50% compared to the 1st order model, highlighting the importance of social data. The regression coefficients show that the added value is mainly attributed to the Twitter data, which is because this variable has a larger correlation with the observed snow. So far we haven't distinguished between image-based or text-based tweets, which is due to the fact that both have a similar effect, i.e.:

$$\hat{Y}_t = 0.17SD(t) + 0.98I_t + 6.07 \tag{6}$$

The question raised here is whether using both Twitter variables Tt and It at the same time would make any sense. We can't assume that these two variables are independent, given that when someone will use text-based tweets is also likely to use image-based tweets for the same reason and vise versa. However, the best criterion on whether both these variables should be used together is the error minimization, as it is measured from the MSE. The resulting regression model is:

$$\hat{Y}_t = 0.16SD(t) + 0.43I_t + 0.11T_t + 3.53 \tag{7}$$

which further reduces the MSE by 10%, as it is shown in Table 1. This indicates that these variables can be used together. We can also see that the contribution of the information from social media is not shared equally, with a much larger contribution from the image-based tweets. This is due to the fact that I_t has a larger correlation with the actual snow, and a smaller weight is given to T_t which only accounts for the added skill to I_t.

For area B, despite the larger overestimation of the observed snow depth during February, the results are qualitatively similar with those of area A. Given the increased correlation of the Sentinel 1 derived snow depth, the MSE exhibits an even larger deflation, compared to area A, as we move from the raw estimate to the different versions of the regression model.

The comparison of the models, presented in Equations (4), (5) and (6) with respect to MSE are presented in Table 9.

Table 9: Mean Square Error (MSE) of observed snow depth with the proposed snow depth estimates.

Snow depth $\hat{Y}$	MSE (Area A)	MSE (Area B)
$\hat{Y}_t = \alpha_1 SD(t)$	2044.80	5656.35
$\hat{Y}_t = \alpha_1 SD(t) + \beta$	142.50	157.85
$\hat{Y}_t = \alpha_1 SD(t) + \alpha_2 T_t + \beta$	76.66	78.39
$\hat{Y}_t = \alpha_1 SD(t) + \alpha_2 I_t + \beta$	71.26	64.98
$\hat{Y}_t = \alpha_1 SD(t) + \alpha_2 I_t + \alpha_3 T_t + \beta$	**67.33**	**64.97**

Our approach aims to complement existing simulated models and not to replace them with social data. Sentinel 1 images, social media textual and visual content are combined to derive a novel model that is able to improve snow depth estimation in urban or near-urban areas. Citizens act as sensors generating multimodal (text-image) data, which can be further utilized to enhance existing state-of-the-art models for snow depth estimation.

Our study, and more particularly the satellite derived snow depth, does have some limitations that must be outlined. The Lievens' method works well for large barren (no trees) areas with deep snow (several meters). Unfortunately, these criteria are not fully met in this study. Our region is close to sea level, with small signal to noise ratio due to the relatively shallow snow (average 20-30cm), not ideal for using C-band satellite backscatter to estimate snow, but there was no alternative for this case. Additionally, the region used to carry out this study was relatively small (a total of 200Km^2), and due to the extensive forest coverage, that strongly affects backscatter, roughly 50% of it had to be dismissed from the analysis making it even smaller for our purposes. Using Sentinel 1 images that cover a larger area would allow an even larger noise reduction. However, the presence of the Twitter data does allow some improvements when satellite-based estimations are combined with relevant tweets and images that contain snow.

Furthermore, the algorithm implemented to rescale the snow index into snow depth, was optimized (coefficients α and β) based on observations of deep snow from barren mountainous regions over the entire Northern Hemisphere. Future developments in remote sensing might address some of these issues, which could enhance the use of satellite backscatter in snow depth applications in small areas like ours.

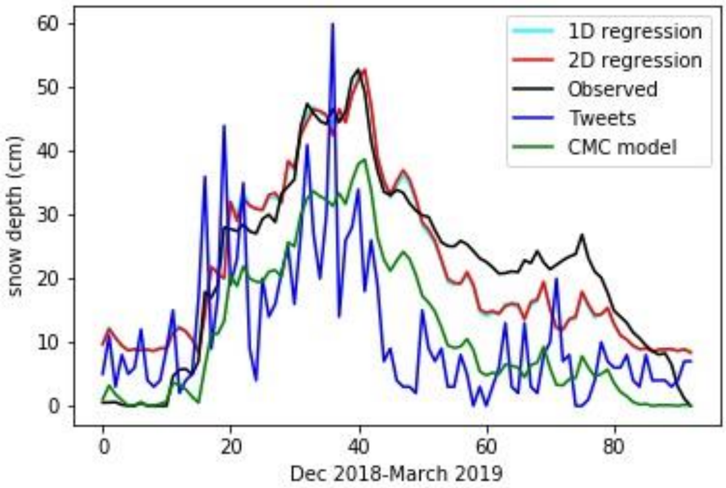

Figure 29: *Variation of observed and estimated snow depth*

One may use daily snow depth simulations, e.g. the ones given by the Canadian Meteorological Centre (CMC), which use in-situ data measurements and sensor measurements from existing infrastructures from weather stations. This would exhibit a strong correlation with the observed snow depth, reaching values up to r = 0.92 (Figure 28). In this case, using a 1D regression model does reduce the MSE to less than a quarter of its original value, reducing the bias during almost the whole period. During days when the snow depth is larger than 25cm the bias is reduced to less than a few centimeters. Examining whether the social media data can improve this, it turns out that introducing the Twitter data in the 2D regression model does not have any effect on the snow depth estimate, and the resulting curve almost coincides with the one from the 1D model. In this case the model would be of the form:

$$\hat{Y}_t = 1.17SD^{simul} - 0.01T_t + 9.04 \quad (8)$$

This is because the correlation of the simulated and the observed snow depth is very high, and feeding additional information to the regression model does not add any new estimation capacity, hence the variable representing the Twitter data is represented by a coefficient that is close to zero.

As expected, social media data from citizen observations are utilized only for complementary purposes when sensor networks do not exist, taking into account that the social media data are openly available and free, and do not require system maintenance costs.

5 CONCLUSIONS

In this deliverable, we have presented the similarity retrieval for EO, non-EO data by considering single modalities, fusion methods that consider the intrinsic characteristics of the data and features produced after applying textual and visual analysis techniques. Also, a fusion approach that considers both EO and non-EO for estimating snow depth was presented.

Starting with the similarity retrieval in non-EO data, we compared the performance of single modalities, i.e. text, visual features, visual concepts, geolocation and time in the similarity retrieval task and the performance of several late fusion methods, including one that has been proposed within EOPEN. Given, that the datasets used for evaluating these methods were significantly large it wasn't possible to annotate them and thus the evaluation was limited to *top-10* results per modality. The results showed that the proposed fusion method outperformed the single modalities and the Condorcet and Reciprocal fusion methods, while it has similar performance compared to Borda fusion when the order of retrieved results is not considered in the top-10 results.

As far the similarity retrieval in EO data, we compared the performance of various approaches in the fields of visual content similarity and similarity by content. The well-known pretrained networks proved useful at the extraction of features, with VGG19 achieving the best scores. At the same time the custom DNN failed to follow the performance of the VGG. Things seem to change to the concept extraction task. The Inception-ResNet v2 provides best results on par with the custom 5 channel DNN. Also, the significance of increasing from the 3 channels (classic RGB) to 5 channel images is evident in this case study. Moreover, we used the previous modalities paired with these of geolocation and time in order to form a late fusion method that improves the retrieved results. Finally, the results demonstrated the importance of combining multiple modalities of an image, managing to lead the board of metrics among 3 other known ranking methodologies.

Finally, we evaluated the potential for using space-borne SAR backscatter measurements to estimate snow depth in areas that experience a shallower snow and are much more densely forested, and how we can combine these with instant social media data to augment our estimate. Our results show that despite the limitations of the backscatter methodology to estimate snow depth, significant improvement can be achieved through the use of regression analysis and social media data. At first, the linear relationship between the snow depth estimate and the observed snow, allows the removal of most of the bias and the noise from the rough estimate. To a second degree a fusion of Twitter data into the regression model allows an additional bias removal. This approach shows that this fusion of social data and Sentinel images has a strong potential.

6 REFERENCES

Bernier, M., Fortin, J.-P., Gauthier, Y., Gauthier, R., Roy, R., Vincent, P., 1999. *“Determination of snow water equivalent using radarsat SAR data in eastern Canada”*, Hydrological Processes 13(18), 3041–3051.

Borda, J. D. 1784. *“Mémoire sur les élections au scrutiny”*, Histoire de l'Academie Royale des Sciences pour 1781.

Cervone, G., Sava, E., Huang, Q., Schnebele, E., Harrison, J., Waters, N., 2016. *“Using twitter for tasking remote-sensing data collection and damage assessment: 2013 boulder flood case study”*, International Journal of Remote Sensing 37(1), 100–124.

Cormack, G. V., Clarke, C. L., & Buettcher, S. 2009. *“Reciprocal rank fusion outperforms condorcet and individual rank learning methods”*, In *Proceedings of the 32nd international* ACM SIGIR conference on Research and development in information retrieval (pp. 758-759).

de Bruijn, J. A., de Moel, H., Jongman, B., de Ruiter, M. C., Wagemaker, J., Aerts, J. C., 2019. *“A global database of historic and real-time flood events based on social media”*, Scientific Data 6(1), 1–12.

Gialampoukidis, I., Liparas, D., Vrochidis S., and Kompatsiaris, I. 2016e. “*Query-based topic detection using concepts and named entities*”, Proceedings of the 1st International Workshop on Multimodal Media Data Analytics (MMDA 2016), pp. 9–13.

Gialampoukidis, I., Moumtzidou, A., Tsikrika, T., Vrochidis, S., andKompatsiaris, I. 2016a. “*Retrieval of multimedia objects by fusing multiple modalities*”, Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (pp.359-362), ACM.

Gialampoukidis, I., Tsikrika, T., Vrochidis, S., andKompatsiaris, I. 2016b. “*Community detection in complex networks based on DBSCAN* and a Martingale process*”, Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11thInternational Workshop on (pp. 1-6), IEEE.

Gurajala, S., Dhaniyala, S., Matthews, J. N, 2019, *“Understanding public response to air quality using tweet analysis”*, Social Media+ Society 5(3), 2056305119867656.

Heverin, T., Zach, L., 2012, *“Use of microblogging for collective sense-making during violent crises: A study of three campus shootings”*, Journal of the American Society for Information Science and Technology 63(1), 34–47.

Jégou, H., Douze, M., Schmid, C. 2011. *"Product quantization for nearest neighbor search"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, pp. 117-128.

Jégou, H., Douze, M., Schmid, C., & Pérez, P. 2010. "Aggregating local descriptors into a compact image representation", In 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 3304-3311). IEEE.

Kendra, J. R., Sarabandi, K., Ulaby, F. T., 1998. *“Radar measurements of snow: Experiment and analysis”*, IEEE Transactions on Geoscience and Remote Sensing 36(3), 864–879.

Lepy, E., Pasanen, L., 2017. *“Observed regional climate variability during the last 50 years in reindeer herding cooperatives of Finnish fell Lapland”*, Climate 5 (4), 81.

Li, J., Rao, H. R., 2010. *“Twitter as a rapid response news service: An exploration in the context of the 2008 china earthquake”*, The Electronic Journal of Information Systems in Developing Countries 42 (1), 1–22.

Li, Y., Zhang, Y., Tao, C., & Zhu, H., 2016. *“Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion”. Remote Sensing*, *8*(9), pp. 709.

Lievens, H., Demuzere, M., Marshall, H.-P., Reichle, R. H., Brucker, L., Brangers, I., de Rosnay, P., Dumont, M., Girotto, M., Immerzeel, W. W., 2019. *“Snow depth variability in the northern hemisphere mountains observed from space”*, Nature communications 10(1), 1–12.

Liu, Y., Chen, C., Han, Z., Ding, L., & Liu, Y. 2020. *“High-Resolution Remote Sensing Image Retrieval Based on Classification-Similarity Networks and Double Fusion”*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp 1119-1133.

Liu, Y., Liu, Y., & Ding, L., 2017. *“Scene classification based on two-stage deep feature fusion",* IEEE Geoscience and Remote Sensing Letters, 15(2), pp 183-186.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. 2013. “*Distributed representations of words and phrases and their compositionality”*, In Advances in neural information processing systems (pp. 3111-3119).

Montague, M., & Aslam, J. A. 2002. *“Condorcet fusion for improved retrieval”*, In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 538-548).

Moumtzidou, A., Andreadis, S., Markatopoulou, F., Galanopoulos, D., Gialampoukidis, I., Vrochidis, S., Mezaris, V., Kompatsiaris, I., Patras, I. 2018. “*VERGE IN VBS 2018”*, In 24th International Conference on Multimedia Modeling (pp. 444-450), Bangkok, Thailand. Springer, Cham.

Nagler, T., Rott, H., Ripper, E., Bippus, G., Hetzenecker, M., 2016. *“Advancements for snowmelt monitoring by means of sentinel-1 SAR”*, Remote Sensing 8(4), 348.

Nagler, T., Rott, H., Schwaizer, G., Ossowska, J., Nemec, J., Fasching, U., 2018. *“Operational monitoring of alpine snow cover within the European Copernicus programme”*, in: Proceedings of the International Snow Science Workshop, Vol. 1, pp. 348–252.

Oh, O., Kwon, K. H., Rao, H. R., 2010. *“An exploration of social media in extreme events: Rumor theory and twitter during the Haiti earthquake 2010”*, in: Icis, Vol. 231, pp. 7332–7336.

Pennington, J., Socher, R., & Manning, C. D. 2014. *“Glove: Global vectors for word representation”*, In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Pittaras, N., Markatopoulou, F., Mezaris, V., Patras, I. 2017. *“Comparison of Fine-tuning and Extension Strategies for Deep Convolutional Neural Networks”*, Proc. 23rd Int. Conf. on MultiMedia Modeling (MMM'17), Reykjavik, Iceland, Springer LNCS vol. 10132, pp. 102-114.

Shi, J., and Dozier, J., 2000. *“Estimation of snow water equivalence using sir-c/x-sar. ii. inferring snow depth and particle size”*, IEEE Transactions on Geoscience and Remote sensing 38(6), 2475–2488.

Strozzi, T., Matzler, C., 1998. *“Backscattering measurements of alpine snowcovers at 5.3 and 35 GHz”*, IEEE Transactions on Geoscience and Remote Sensing 36(3), 838–848.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. *“Going deeper with convolutions”*, In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Tang, X., & Jiao, L., 2016. *“Fusion similarity-based reranking for SAR image retrieval”*, IEEE Geoscience and Remote Sensing Letters, 14(2), pp. 242-246.

Turunen, M. T., Rasmus, S., Bavay, M., Ruosteenoja, K., Heiskanen, J., 2016. *“Coping with di cult weather and snow conditions: Reindeer herders’ views on climate change impacts and coping strategies”*, Climate Risk Management 11, 15–36.

Vijaymeena, M. K., and Kavitha, K., 2016. “*A Survey on Similarity Measures in Text Mining”*, Machine Learning and Applications: An International Journal, 3 (1), 19-28.

Wang, Z., Ye, X., Tsou, M.-H., 2016. *“Spatial, temporal, and content analysis of twitter for wildfire hazards”*, Natural Hazards 83(1), 523–540.

Yan, J., 2009. *“Text Representation”*, Encyclopedia of Database Systems, 3069-3072

A Appendix

A.1. Retrieval Results for Non-EO Content

In the following, we present for each use case the *top-10* results returned for each single modality and for the four different methods, i.e. the proposed fusion algorithm, the Borda fusion algorithm, the Reciprocal fusion algorithm, and the Condorcet fusion algorithm.

A.1.1 Flood Use Case

Figure 30: *top-10* similar results retrieved using Text and Time modalities.

Figure 31: *top-10* similar results retrieved using Visual Features and Visual Concepts modalities.

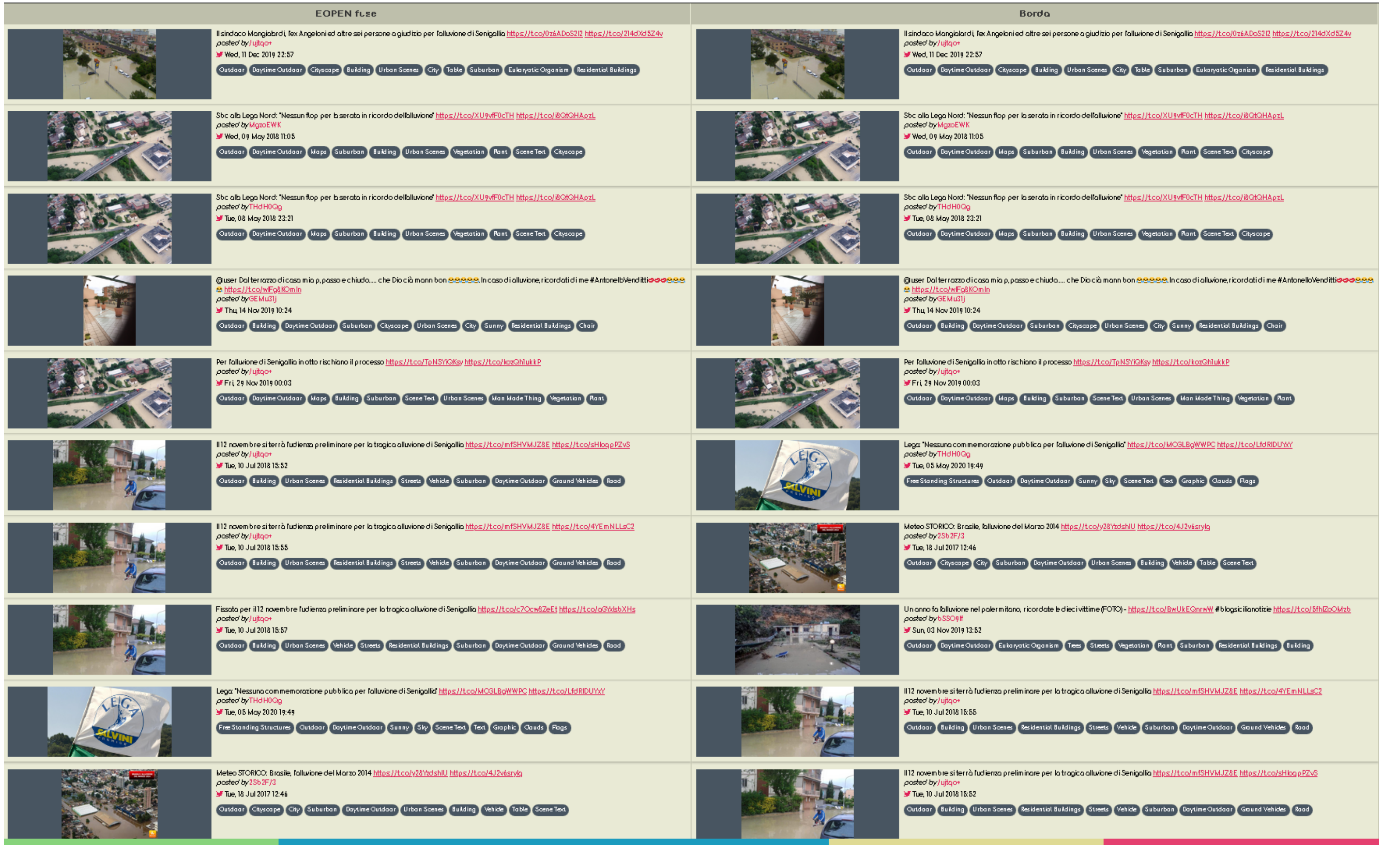

Figure 32: *top-10* similar results retrieved using EOPEN and Borda fusion algorithms.

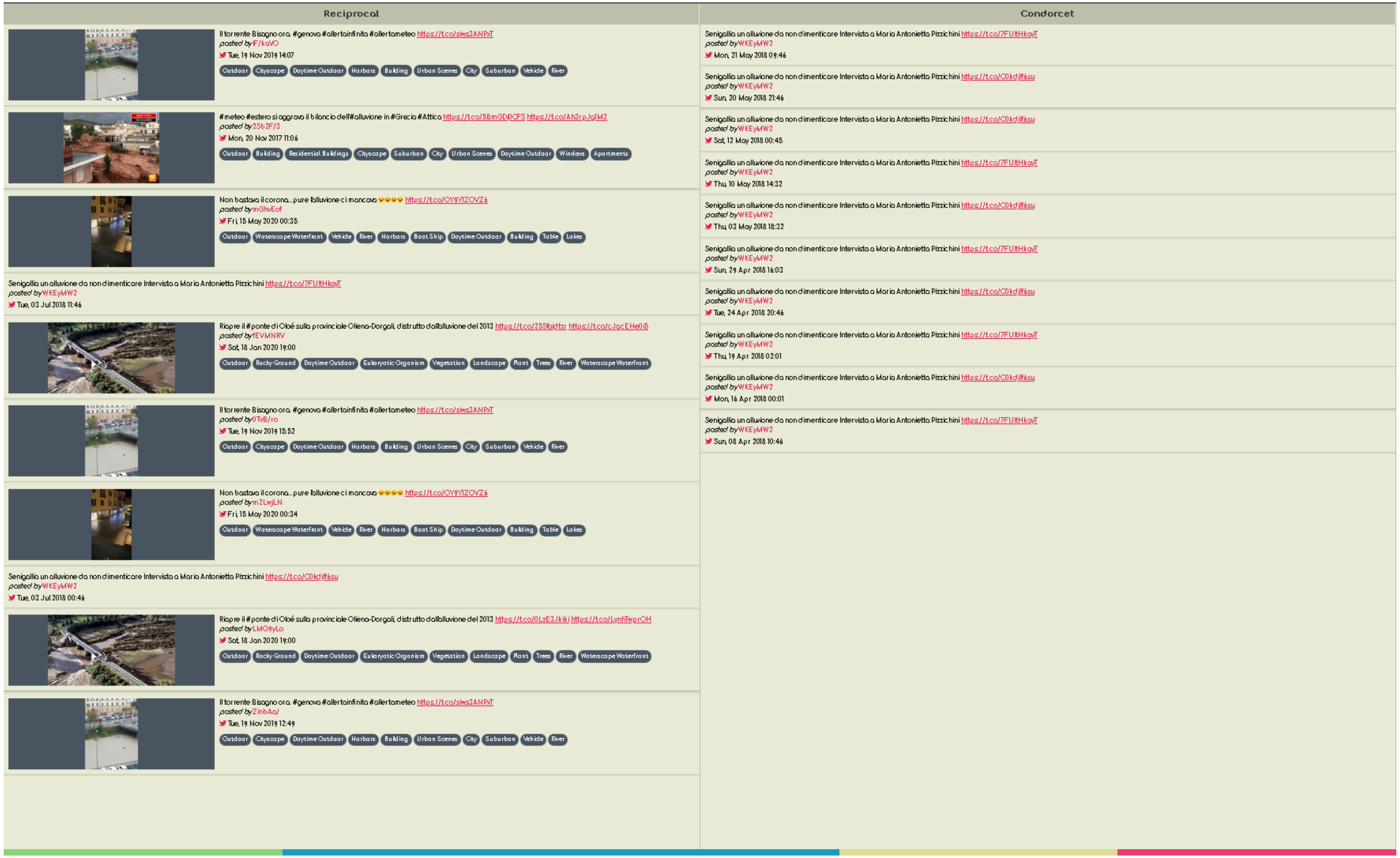

Figure 33: *top-10* similar results retrieved using Reciprocal and Condorcet fusion algorithms.

A.1.2 Food Use Case

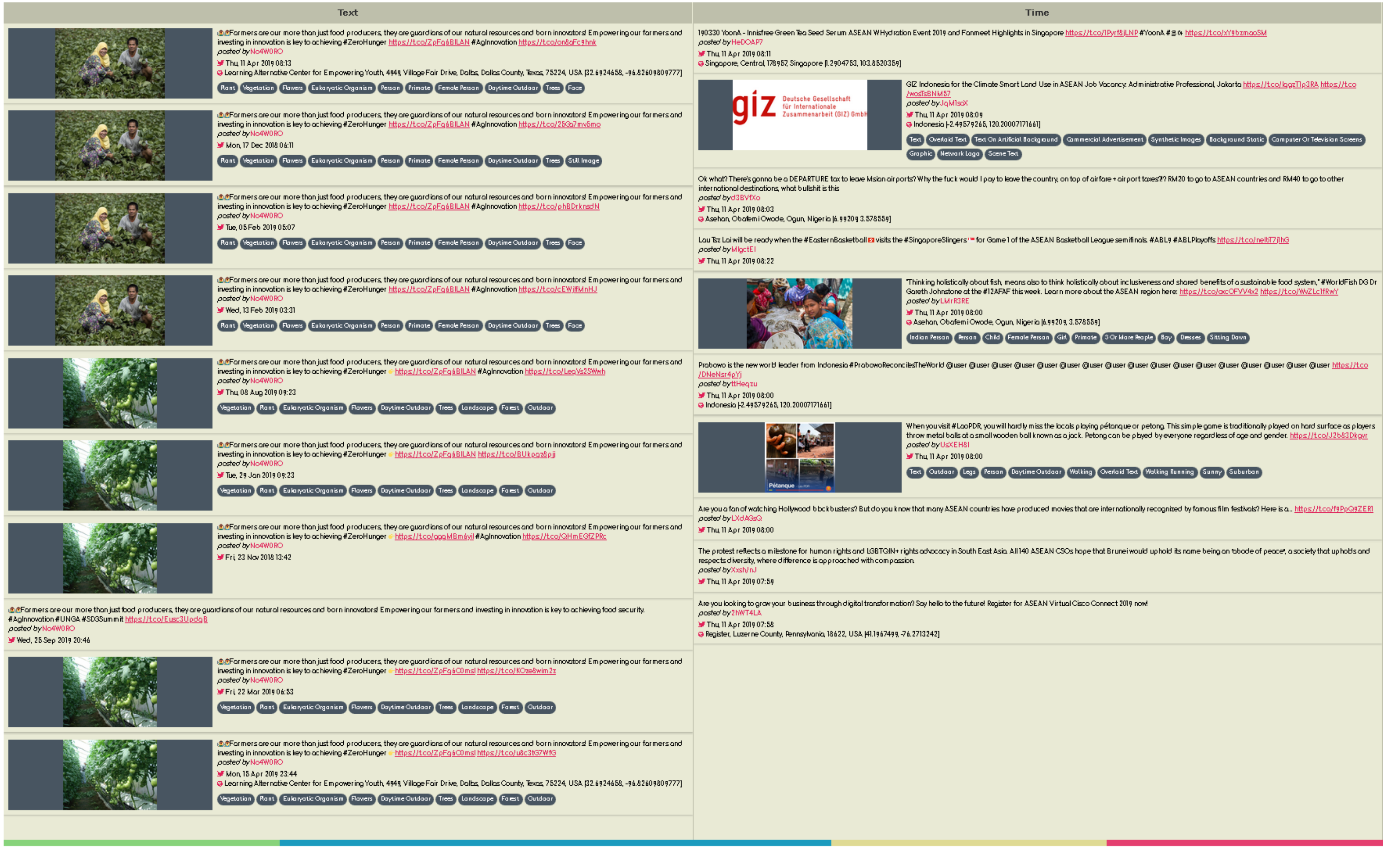

Figure 34: *top-10* similar results retrieved using Text and Time modalities.

Figure 35: *top-10* similar results retrieved using Visual Features and Visual Concepts modalities.

Figure 36: *top-10* similar results retrieved using EOPEN and Borda fusion algorithms.

Figure 37: *top-10* similar results retrieved using Reciprocal and Condorcet fusion algorithms.

A.1.3 Snow Use Case

Figure 38: *top-10* similar results retrieved using Text and Time modalities.

Figure 39: *top-10* similar results retrieved using Visual Features and Visual Concepts modalities.

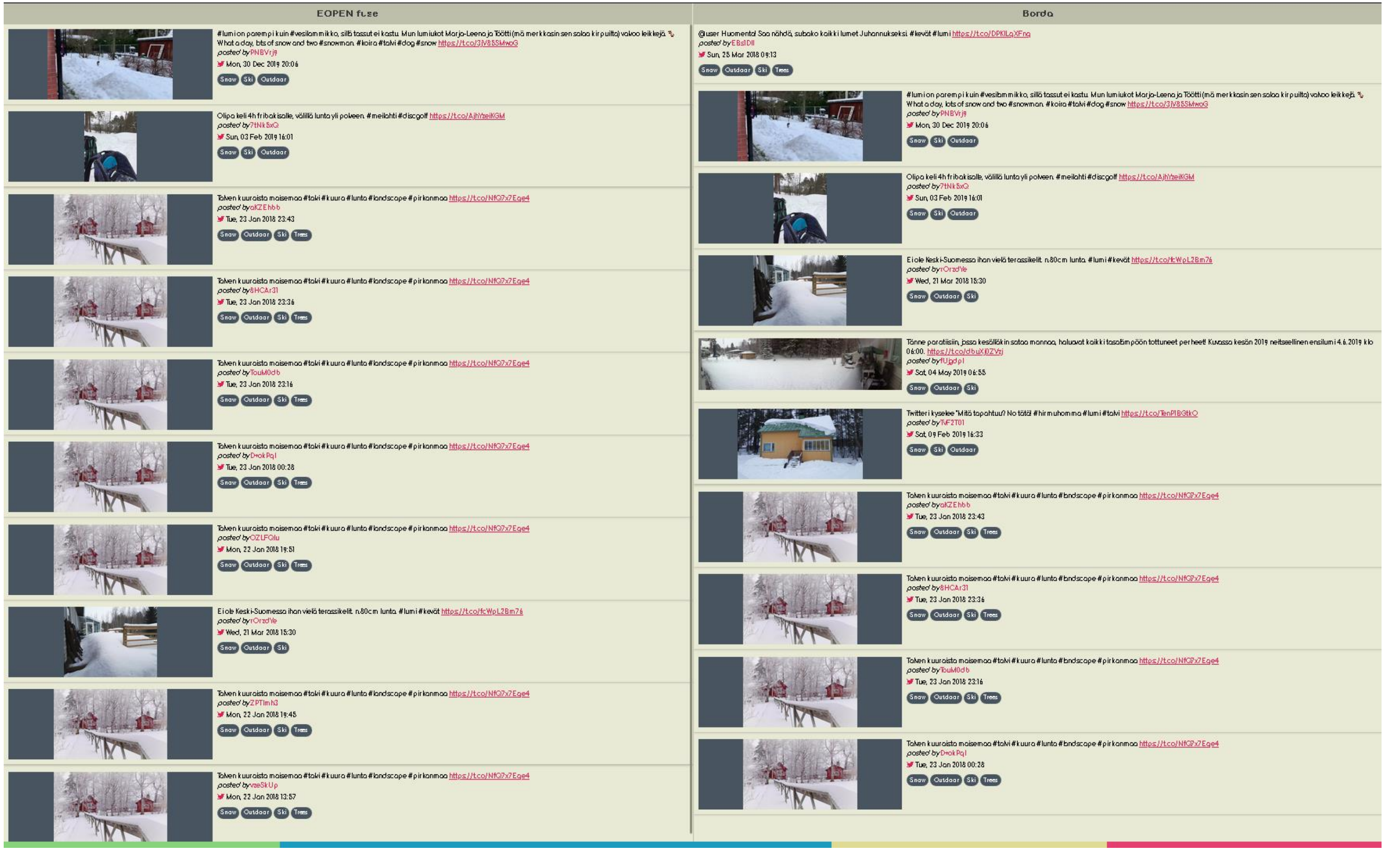

Figure 40: *top-10* similar results retrieved using EOPEN and Borda fusion algorithms.

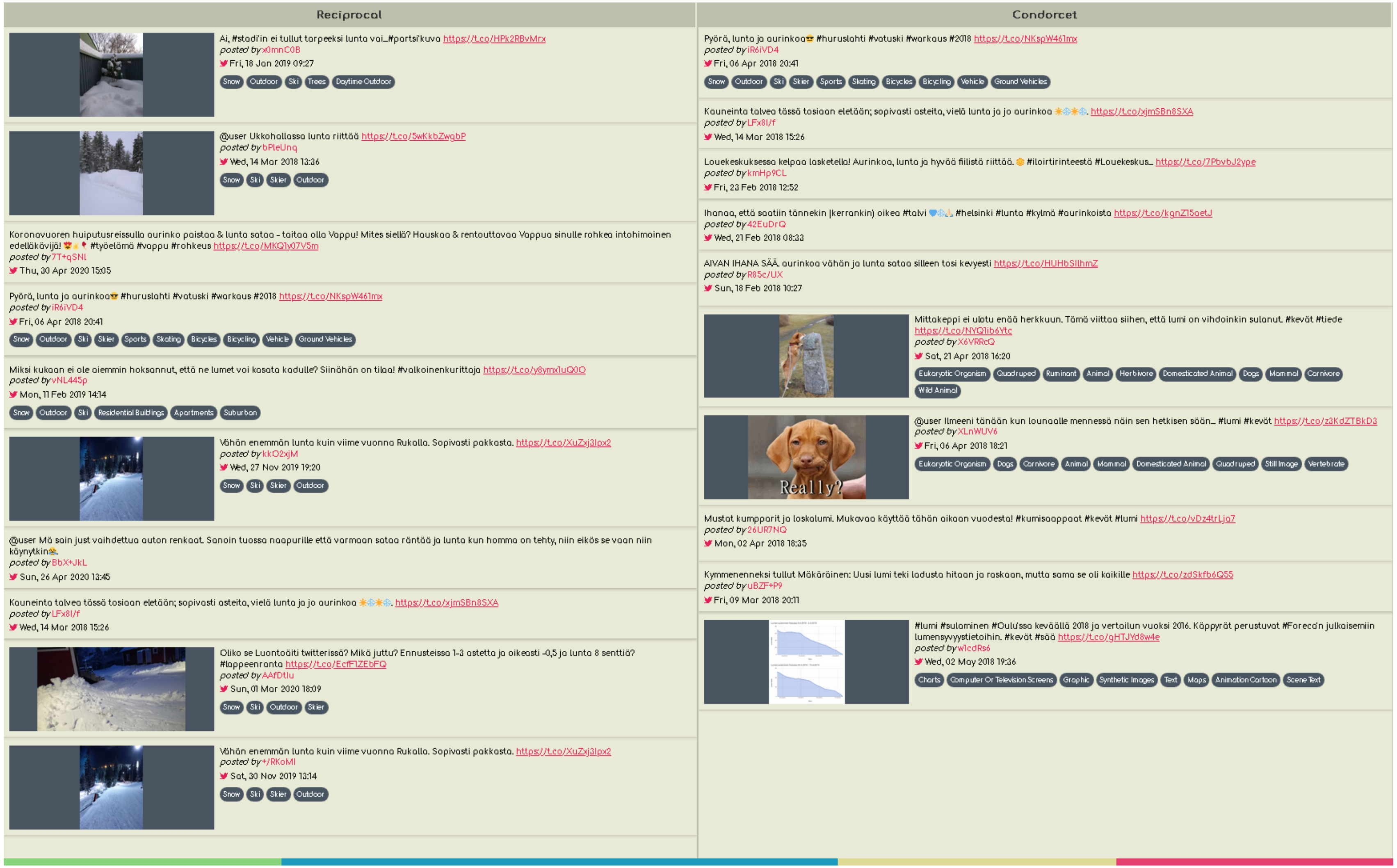

Figure 41: *top-10* similar results retrieved using Reciprocal and Condorcet fusion algorithms.

A.2. Information Retrieval in EOPEN and CANDELA

Information Retrieval (CERTH-DLR)	EOPEN EO data	EOPEN non-EO data	CANDELA
Input data	Sentinel 2 images in GeoTIFF format	Twitter data in JSON format	Sentinel-2 and Sentinel-1 images, other multispectral and SAR sensors too
Query	Sentinel 2 patch (120x120)	One tweet	Image patch of a user-defined dimension, CBIR and semantic queries – concept search
Pre-processing	Multi-band GeoTIFF	Localization of named entities Concept extraction Visual feature extraction Lucene indexing	Spatial overlap between S1 and S2 images, for Data Fusion
Features extracted	• VGG19, fc2 (Dense) layer, 1x4096 • ResNet50, avg_pool (Dense) layer, 1x2048 • Inception-ResNet_v2, avg_pool_layer (Dense) layer, 1x1536	• Text (Apache Lucene indexed) • DCNN visual features (Extended GoogLe net) • TRECVID 345 visual concepts • Timestamp • Location extracted	• Multispectral histograms • Gabor descriptors (MPEG standard) • Weber features (WLD) for multispectral images • Modified Weber features for SAR images
Similarity per modality	• Euclidean distance for the feature and concept vectors,	• Text search (Lucene) • Euclidean distance for the feature and	Active learning based on SVM and Bayesian decision SQL multimodal queries (image

	• Sort by time for the timestamp • Centroid-to-centroid for the location (GEOnear, mongoDB)	concept vectors, • Sort by time for the timestamp • Point-to-point for the location (GEOnear, mongoDB)	semantics, EO product metadata, image descriptors)
Data Fusion	Late fusion on the order of results per modality	Late fusion on the order of results per modality	S 1& S2 data fusion with active learning at image feature level S1 &S2 fusion at semantics via SQL
Supervised/Unsupervised	• The fusion is unsupervised • The feature extraction is supervised	• The fusion is unsupervised • The feature extraction is supervised	• The data mining and fusion are supervised • The feature extraction is unsupervised
Open/closed source	Open source at the end of the project	Open source at the end of the project	Open source
API or GUI available	Will be available as a web service	Will be available as a web service	GUI available
Use of Docker	No	No	Yes
Dataset	BigEarthNet (part of it with specific classes) • Water • Snow • Rice • Forest • Vineyards • Rock	~10,000,000 tweets regarding the EOPEN use cases	Sentinel 1 and 2 semantic annotation in active learning (user in the loop – dialog HMC) user is annotating the retrieved Open number of classes defined by the user adapted to the application (up to 100) Benchmark data sets created

	• Urban		in the project are available In the project have been analyzed up to 1Mkm^2 covered by S2 and S1
Evaluation metrics	Mean Average Precision	Precision@k	Precision and Recall