UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Ground-based Cloud Classification with Deep Learning

Author:
Marcos PLAZA

Supervisor:
Jordi VITRIÀ, Gerard GÓMEZ

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2022

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Ground-based Cloud Classification with Deep Learning

by Marcos PLAZA

In a context of climate change, clouds play an essential role in the circulation of water vapour and affects the earth's energy balance. In the study of weather forecasting clouds are always regarded as the core factor. The classification of the different types of clouds is mainly useful for weather forecasting. As a consequence, we can also characterise the climate of a region and help to explain other phenomena that may be related to the climatology of the area. Weather prediction is an advantage when it comes to taking action in many cases where climate has a significant impact, for example in cases such as air transport.

The long-term objective of this master's thesis is to characterise the climate of the city of Barcelona through the development of a tool to carry out automatic cloud classification from images taken from the ground. These images come from the Observatori Fabra located at one of the highest points of the city, more specifically on the Tibidabo mountain.

For the development of this classification tool we are going to explore different solutions from the field of deep learning and computer vision; mainly convolutional neural networks as well as vision tranformers. Secondly, given that the different data sources, our own (images taken from the Fabra Observatory) and other existing datasets (such as Cirrus Cumulus Stratus Nimbus abreviated as CCSN or Swimcat), contain not too much data to achieve a remarkable result, we have decided to use a model that uses that uses three Siamese neural networks where images will be combined to generate more different input data.

All the code dedicated to solving the problem of cloud classification, is available in this public Github repository.

Acknowledgements

In first place, I want to express my gratitude to all of my supervisors for their outstanding assistance, dedication, and expertise in assisting me with this project.

I would especially like to thank Jordi for patiently guiding me at all times and giving me the tools and the knowledge to overcome the obstacles that arose week after week. Also to Gerard for his support and for being the link of contact with Alfons Puertas – Oservatori Fabra, RECAB.

To Alfons Puertas for providing the magnificent photographs of the clouds, which although they have served to feed our models, have both a great scientific and artistic value.

I would want to express my gratitude to all of the professors who have contributed to the Fundamental Principles of Data Science Master's Program at the University of Barcelona. I wouldn't have developed the knowledge or inspiration for this project without this teaching.

Contents

Chapter 1

Introduction

Over the last decade, due to advances in both hardware and software, the field of artificial intelligence has been able to grow and evolve. As a consequence, computer vision has been able to take advantage of the development of deep learning to obtain astonishing solutions to a wide range of problems; facial and object recognition, crowd counting, generation of three-dimensional graphics from a finite set of two-dimensional images (Martin-Brualla et al., 2020) and even models that are capable of classifying samples never seen before (zero-shot classification) by combining vision with the power of natural language processing (Radford et al., 2021). All these examples are part of the problems solved to a greater or lesser extent through architectures such as convolutional neural networks (or commonly abbreviated CNN), such as those used to solve the problem of classifying clouds from photographs taken from the ground.

Clouds are an indicator of climate and weather on earth. The ensemble of radiating effects can be affected by the presence of different cloud types. Therefore, identifying the type of cloud is important when characterising the climate of a region. Furthermore, the classification of the different types or genres of clouds has the clear objective of predicting the weather of a region slightly in advance. So, what we want to solve on this occasion is a cloud classification problem. Traditional cloud classification or identification relies heavily on the experience of observers and is very time-consuming. We propose to develop a neural network for accurate cloud classification on the ground. To this end, we will explore convolutional neural network architectures, a vision transformer, as well as other models that have greater flexibility from little data.

A cloud is a hydrometeor[1] consisting of minute particles of liquid water or ice, or of both, suspended in the atmosphere and usually not touching the ground. It may also include larger particles of liquid water or ice, as well as non-aqueous liquid or solid particles such as those present in fumes, smoke or dust.

Although, clouds can be classified in various ways, the international classification (according to the World Meteorological Organization) of clouds considers ten basic forms or genera, under the morphological properties of the clouds, which are as follows; Cirrus, Cirrocumulus, Cirrostratus, Altostratus, Altocumulus, Stratus, Stratocumulus, Nimbostratus, Cumulus and Cumulonimbus. Since this type of classification is closely related to the physical constitution of clouds, there are visual characteristics that can be measured from the ground. Therefore, it has been decided to work in the first instance with this type of classification collected in the CCSN (Cirrus Cumulus Stratus Nimbus) dataset, as well as in the dataset that has been built from images taken at the Observatori Fabra. Alternatively, it has been decided

[1] any water or ice particles that have formed in the atmosphere or at the Earth's surface as a result of condensation or sublimation.

to classify clouds into much more simplified physical characteristics, as in the case of Dev, Lee, and Winkler, 2015 where the Swimcat dataset (Singapore Whole-sky Imaging Categories) is used. This database includes the following categories: Clear Sky, Patterned Clouds, Thick White Clouds, Thick Dark Clouds and Veil Clouds. There are several studies based on Swimcat, but the categories are not including all the required cloud categories, so maybe this one is simplest but at the same time is insufficient from the perspective of meteorological research and applications.

In this report you will find a review from the most general meteorological fundamentals of cloud classification. We will then look at the main characteristics of the data used, and review in detail the deep learning models which has been implemented;Convolutional Neural Networs, Vision Transformer and Siamese Networks with triplet loss function. Finally we will see how these models perform against the available data by giving some details of the experimentation and the results. We will solve the problems of accurate classification by supporting the conformal prediction framework, as well as the use of visualisation tools to see which cloud types are most closely related to each other.

Chapter 2

Clouds and cloud classification

The international classification of clouds, starting with the main division of clouds into ten genera (any cloud can be assigned to one and only one of these genera), is a morphological classification, i.e. it distinguishes clouds according to their shape and appearance. However, the shape of clouds is closely related to their genesis, so that the mentioned classification also has a certain genetic character. Apart from the purely scientific interest of classification, cloud classification also has an obvious utility in weather prognosis, at least on small spatial and temporal scales. Thus, careful observation of the cloud types on the visible horizon by an experienced observer should be able to make an acceptable weather forecast in the short term, e.g. a couple of hours, and in a certain area around the horizon, of about 10 kilometres of radius. It will be a forecast that can be used to ensure, most of the time, reasonable weather conditions for the development of an outdoor activity in the indicated time and space.

A cloud is a hydrometeor consisting of tiny particles of liquid or icy water, or both, suspended in the atmosphere and normally not touching the ground. It may also include larger particles of liquid or ice water, as well as non-aqueous liquid or solid particles such as those present in smoke or dust. A cloud is therefore not water vapour, which is invisible, but water particles in liquid and/or solid states.

Hydrometeor means an atmospheric phenomenon, i.e. observed in the atmosphere or at the surface in contact with it, where the essential element is the water. Hydrometeors are the largest group of atmospheric phenomena or meteors, but as we can see in the table 2.1, there are other types of phenomena.

Type	**Basic component**	**Examples**
Hydrometeors	Water	Clouds, rain, snow, dew, etc.
Lithrometeors	Non-aqueous particles	Haze, smoke, sandstorm, etc.
Photometeors	Light	Rainbow, solar halo, lunar corona, etc.
Electrometeors	Electricity	Thunderstorm, lightning, St. telmo's fire, etc.

TABLE 2.1: Basic clasification of meteors.

Clouds originate from condensation and sublimation of water vapour in the air, which gives rise to liquid droplets and watery ice crystals respectively, most of them are microscopic. The processes of condensation and sublimation of water vapour require air saturation. In the free atmosphere, the most common process to reach

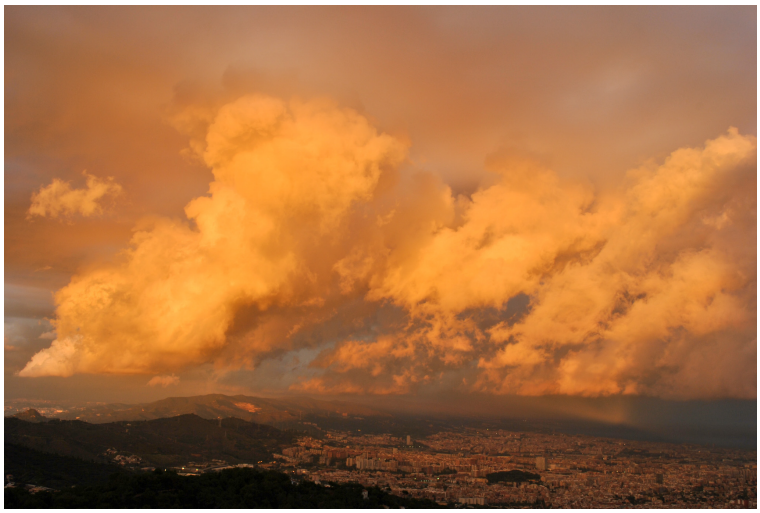


FIGURE 2.1: Cloud picture taken from Observatori Fabra in Barcelona. By Alfons Puertas.

saturation of the air is its cooling by rising air. Secondarily, the horizontal displacement and mixing of air of different temperatures and humidity can also lead to saturated mixing and the formation of clouds and fogs. Air lifts, which cause most clouds, can be classified into four types: a) convective, b) orographic, c) cyclonic and d) frontal. The convective updrafts, for example, are thermal in nature, due to the rising of warm and lightweight air in contact with a warm surface.

2.1 Cloud classification

Cloud classification can be made according to various criteria:

- physical constitution,
- development,
- height and the relationship between vertical dimension and horizontal extension

At the same time, clouds are classified according to their physical constitution as: a) liquid, b) ice crystals, c) icy and d) mixed. Liquid clouds are those composed exclusively of liquid droplets. Ice crystal clouds consist only of these solid particles. Icy clouds (which are rare to see) are made up of frozen water droplets. Mixed clouds are those with a mixture of some or all of the three aqueous elements mentioned above.

Clouds can be classified according to their evolution as: a) local and b) migrating. This classification is of little relevance. A cloud is local if its "life", from its appearance to its dissipation, takes place within sight of a fixed observer on the earth's surface. Otherwise, we speak of an emigrating cloud, such as those which appear at one point on the horizon and, after crossing the sky, disappear at the opposite point or dissipate, or which appear in the view of a fixed observer but disappear over the horizon before dissipating.

Clouds are classified according to their height as: a) high, b) medium and c) low. Low clouds are those located at the low level, which for mid-latitudes or temperate latitudes is the level between the earth's surface and 2 km above sea level. Medium

clouds are those found at the middle level, between 2 and 7 km above sea level. And high clouds, located at the upper level or upper floor, are between 5 and 13 km high. As can be seen, the middle and upper levels overlap, as there are some cloud genera that sometimes exceed their characteristic level slightly. At polar latitudes, the low, middle and high levels are between the land surface and 2 km, between 2 and 4 km, and between 3 and 8 km, respectively; and at tropical latitudes, between the land surface and 2 km, between 2 and 8 km, and between 6 and 18 km, respectively.

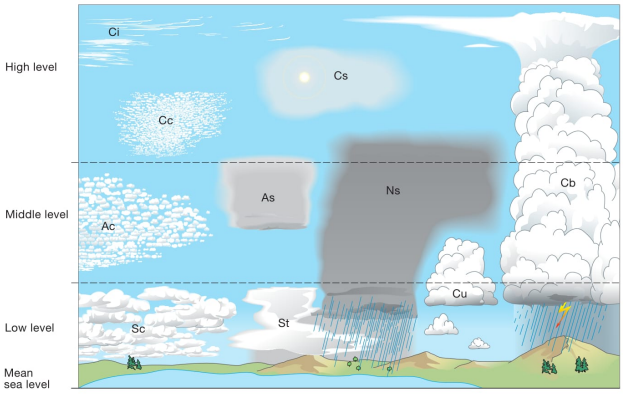


FIGURE 2.2: Cloud genres location according to their height.

The classification of clouds based on the relationship between vertical dimension and horizontal extent consists of two categories: a) stratiform and b) cumuliform. The vertical dimension of a cloud is the vertical distance between its base and its top. The horizontal extent can be defined as the maximum distance between two points of the cloud's projection on the ground. In terms of the relationship between these two distances, a stratiform cloud is one whose vertical dimension, although it may be considerable, is very small in relation to its horizontal extent. A cumuliform cloud, on the other hand, has a vertical dimension comparable to or even greater than its horizontal extent.

Stratiform **Cumuliform**

FIGURE 2.3: Stratiform versus cumuliform

2.2 International cloud classification

Clouds are continuously evolving, often ephemeral, and show an infinite variety of shapes. Moreover, it can be argued that no two clouds are identical. There are, however, a limited number of characteristic shapes, observed worldwide, which allow clouds to be grouped or classified. Specifically, the international cloud classification considers ten basic forms or genera, which form the backbone of the classification. As it is mentioned previously, any cloud observed can be assigned to one and only one of these genera. Species and varieties are then distinguished, as well as other characteristics. This classification system is similar to the taxonomic classification of plants and animals, and similarly uses Latin names, although there are also equivalent names in spoken languages.

The intermediate forms between two genera or in transition between them are often observed. So this fact adds more difficulty in order to classify one cloud in one of the ten genera. In the table 2.2 are some useful descriptions in order to know the main features about them.

Abbreviation	**Name**	**Description**
Ci	Cirrus	Fibrous, white feathery clouds of ice crystals
Cs	Cirrostratus	Milky, translucent cloud veil of ice crystals
Cc	Cirrocumulus	Fleecy cloud, cloud banks of small, white flakes
Ac	Altocumulus	Grey cloud bundles, compound like rough fleecy cloud
As	Altostratus	Dense, gray layer cloud, often even and opaque
Cu	Cumulus	Heap clouds with flat bases in the middle or lower level
Cb	Cumulonimbus	Middle or lower cloud level thundercloud
Ns	Nimbostratus	Rain cloud; grey, dark layer cloud, indistinct outlines
Sc	Stratocumulus	Rollers or banks of compound dark gray layer cloud
St	Stratus	Low layer cloud, causes fog or fine precipitation
Ct	Contrails	Line-shaped clouds produced by aircraft engine exhausts

TABLE 2.2: Summary of cloud genres.

Chapter 3

About data

Expertly labeled data must be available and accessible in order to train any deep learning model that calls for supervised classification tasks. We will see that one of the problems that needs to be solved is the lack of adequate training samples. As this project is intended to work in particular at the Observatori Fabra in Barcelona, thanks to the collaboration of the physicist Alfons Puertas, we have a small sample of the sky in Barcelona. The most well-known and complete data collection, ImageNet (Deng et al., 2009), for instance, contains over 10 million tagged images and is appropriate for various image classification techniques. The ImageNet data will be used in order to implement a pretrained Convolutional Neural Network model (specifically a CNN known as MobileNetV2 (Sandler et al., 2018) which has prove to give excellent results in a wide variety of classification problems by using ImageNet weights).

Otherwise, the intended results are specifically, from cloud images taken from the ground. First of all, as already mentioned above, for the identification of the genera we will make use of the CCSN dataset, while to distinguish between fewer categories and simplify the problem for the recognition of the physical constitution of the clouds we will use the Swimcat dataset. Additionally we have used the dataset baptised with the name Observatori Fabra Clouds and Observatory Fabra Swimcat, in order to carry out both classifications above. Now, it is convenient to provide details and information about the datasets used in this project.

3.1 Cirrus Cumulus Stratus Nimbus (CCSN) dataset

The CCSN dataset contains 2543 cloud images. All images have a fixed resolution of 256×256 pixels in JPEG format. According to the genre-based classification recommendation of the World Meteorological Organization, we divide them into eleven different categories or genres already mentioned (additionally we add the class 'Contrails – Ct' which are the trail that some air vehicles leave in the air, but it is not a type of cloud itself); Ac, Sc, Ns, Cu, Ci, Cc, Cb, As, Ct, Cs, St.

Category	#images
Ci	139
Cs	287
Cc	268
Ac	221
As	188
Cu	182
Cb	242
Ns	274
Sc	340
St	202
Ct	200

TABLE 3.1: Distribution of the CCSN dataset.

3.2 Singapore Whole-sky Imaging Categories (SWIMCAT) dataset

The Swimcat database, contains 784 sky/cloud patch images with 125 × 125 pixels. All images were captured in Singapore using WAHRSIS, a calibrated ground-based whole sky imager, over a period of 17 months from January 2013 to May 2014. The dataset is divided into five distinct categories: Clear Sky, Patterned Clouds, Thick Dark Clouds, Thick White Clouds, and Veil Clouds. There are several studies based on the Swimcat dataset, but maybe in this case the dataset does not include all the required cloud categories, which is insufficient from the perspective of meteorological research and applications.

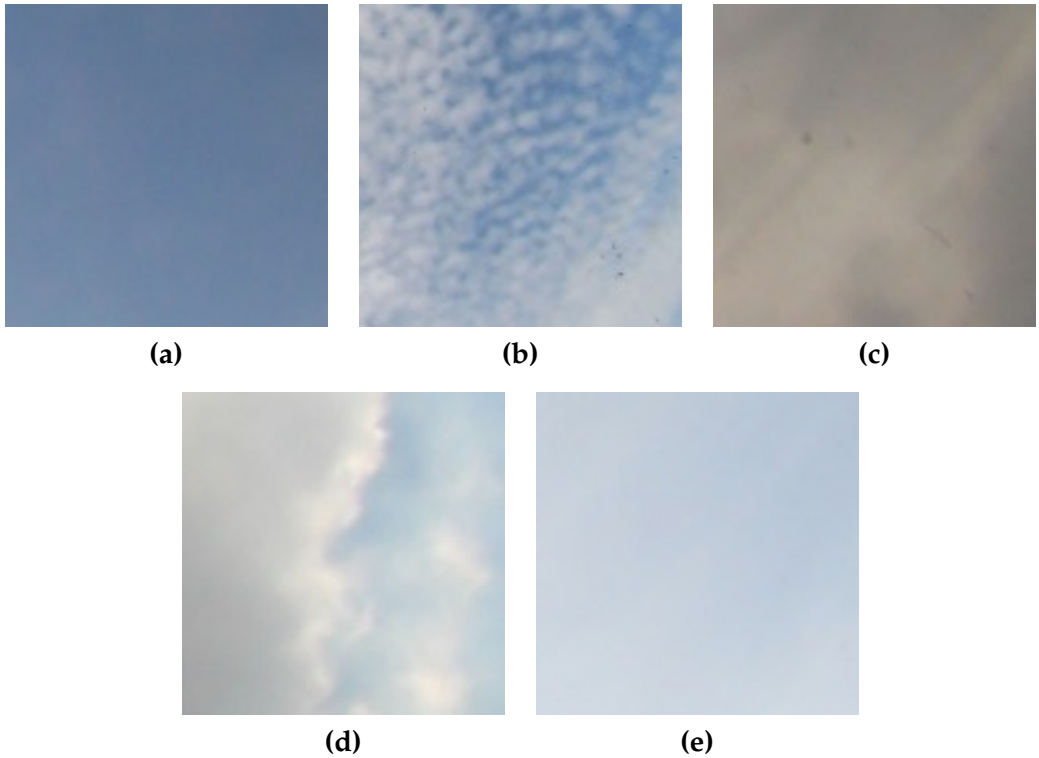


FIGURE 3.1: **(a)** Clear Sky **(b)** Patterned Clouds **(c)** Thick Dark Clouds **(d)** Thick White Clouds **(e)** Veil Clouds

3.3 Observatori Fabra Clouds dataset (FabraClouds)

This dataset has been constructed from photographs taken from the ground by the physicist, meteorologist and photographer, Alfons Puertas – Observatori Fabra, RECAB. He is a very good connoisseur of the sky and has a publication called *"Atlas de núvols de l'observatori Fabra"* where he explains and exposes in an excellent way a great knowledge of the meteorology and a big sample of the sky in the city of Barcelona. The set of 2228 images used, all in RGB codification, have been appropriately treated for cropping and resizing (square images have been used, usually with a size of 128 × 128 pixels or 256 × 256). Due to Alfons' expertise in the field, he has been able to classify the clouds and label them appropriately to make use of them in our deep learning models. On the other hand, it makes much more sense to use images from the observatory itself, as the models are intended to work at this particular location.

FIGURE 3.2: Cumulonimbus over the city of Barcelona, by Alfons Puertas.

However, the main drawback of this dataset is that it does not have a similar number of examples of each class, but maybe with the right techniques and data augmentation we can mitigate the effect of this imbalance.

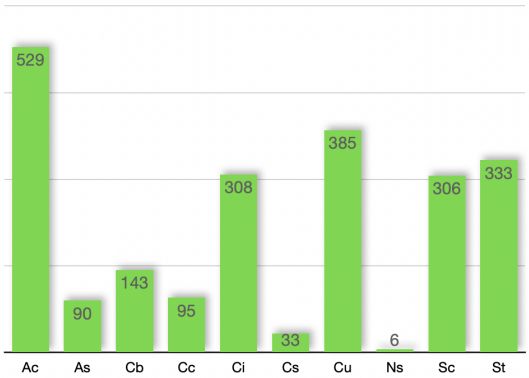


FIGURE 3.3: Distribution of our own dataset named Fabra Clouds.

3.4 Observatori Fabra Swimcat dataset (FabraSwimcat)

This dataset has been created from the photographies provided by Alfons Puertas. After being processed and cropped, the photographs have been reviewed and manually distributed into the five categories of the Swimcat dataset, i.e. Clear Sky, Patterned Clouds, Thick White Clouds, Thick Dark Clouds and Veil Clouds. This database contains 973 RGB images in total.

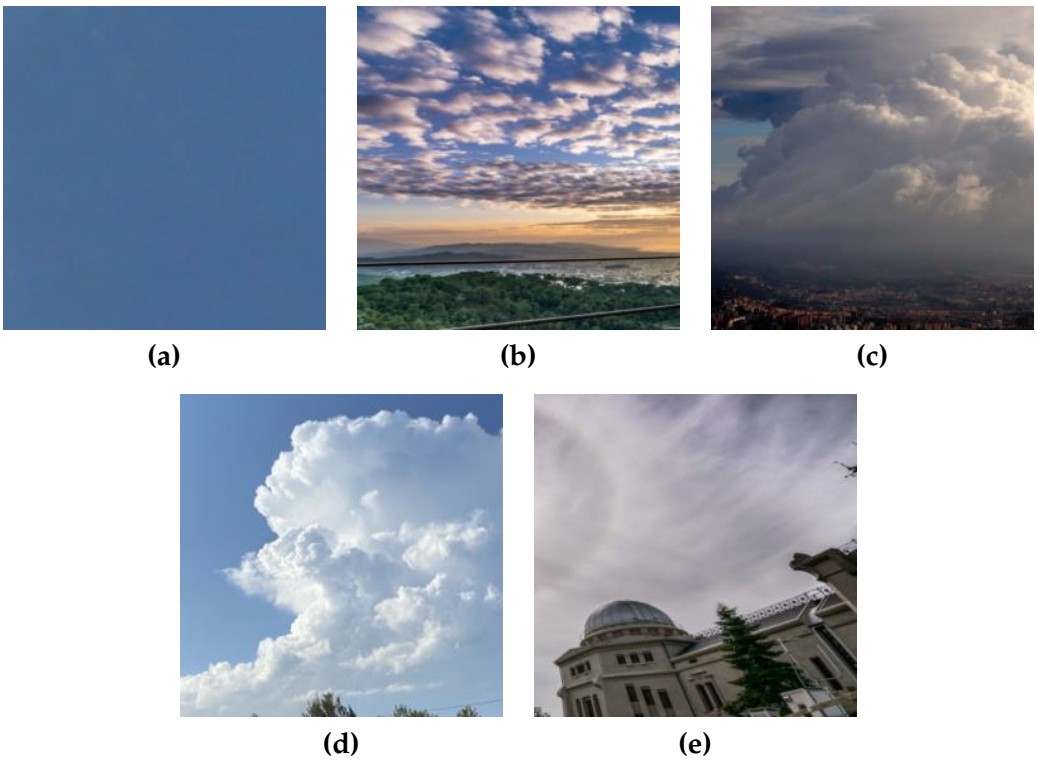


FIGURE 3.4: **(a)** Clear Sky **(b)** Patterned Clouds **(c)** Thick Dark Clouds **(d)** Thick White Clouds **(e)** Veil Clouds

Chapter 4

Methods and techniques

In this section we present the set of deep learning algorithms we have used to achieve the objective of classifying clouds. Initially, we wanted to implement a convolutional neural network to obtain a set of features that differentiate one type of cloud from another. In addition, it was also intended to see how a particular vision transformer architecture would work. As mentioned in the introduction, with the current data available it is hard to get the enough number of images to train and obtain an accurate classifier through these two models. However, while trying to solve this problem, it has been decided to use an architecture that employs three convolutional neural networks in parallel and that uses triplets of images as input. In this way the combination of triplets that can be formed from the available classes is much larger and we are not limited to using data augmentation techniques. Let's dive into every one of them.

4.1 Convolutional Neural Networks (CNNs)

Neural networks are a subset of machine learning, and they are at the core of deep learning algorithms. They are made up of node levels, each of which includes an input layer, one or more hidden layers, and an output layer. Neural networks learn through an algorithm known as backpropagation. This involves comparing the output a network produces with the output it was meant to produce, and using the difference between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input units. In time, backpropagation causes the network to learn, reducing the difference (through an optimization process) between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should.

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are mainly these three:

- Convolutional layers,
- Pooling layers,
- and Fully-connected (FC) layers.

The central component of a CNN is the convolutional layer, which is also where the majority of computation takes place. It needs input data, a filter, and a feature map, among other things. Assume that the input will be a color image that is composed of a 3D pixel matrix. As a result, the input will have three dimensions; height, width, and depth. Additionally, we have a feature detector, also referred to as a

kernel or filter, which will move through the image's receptive fields and determine whether the feature is there. This process is known as a convolution. As previously mentioned, to train such algorithms, or what is the same, to learn the a set of features from the images, we need a dataset big enough, and we need the data to be well labelled and previously classified. This features learned through convolution operations, will be used as pattern detectors in the image, so that we will be able to classify things; this time clouds.

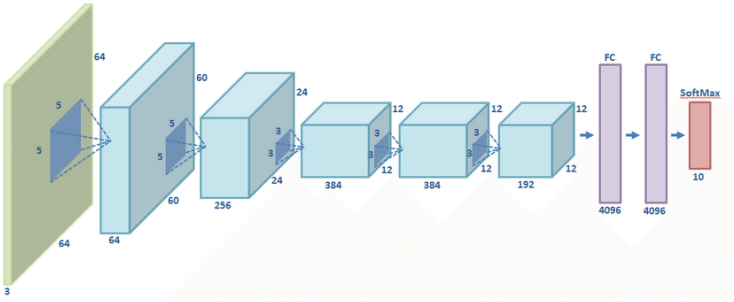


FIGURE 4.1: AlexNet structure scheme.

AlexNet is the name of a convolutional neural network (CNN) particular architecture, designed by Alex Krizhevsky (Krizhevsky, Sutskever, and Hinton, 2012). AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012. FabraCloudNet is an adaptation of the classical AlexNet architecture. It has the following structure.

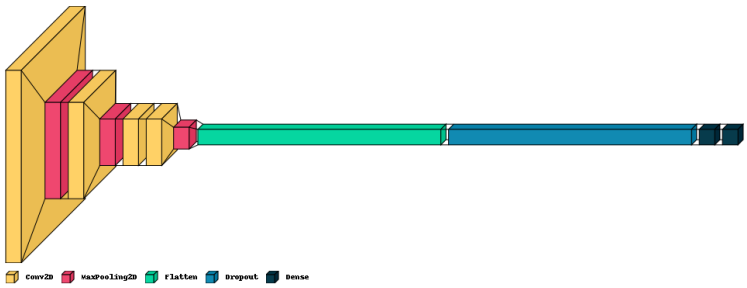


FIGURE 4.2: FabraCloudNet general structure scheme.

As shown in 4.2, the FabraCloudNet consists of four convolutional layers and two fully connected layers. The third and fourth convolutional layers are connected directly without a pooling layer. All this upper layers use the ReLU activation function, that is $max(0, z)$ where z is the value of the output of the neuron. We also apply dropout before the fifth and sixth FC layer. The final layer has the same number of units as the number of classes. It uses the softmax activation function as it produces a distribution output probability (always outputs a value between 0 and 1) for each category. The loss function is another important element in all machine learning algorithms. The error is what we want to minimize. In this case, we used the categorical crossentropy loss function. It is well known for being used in multiclass classification problem. For the optimization algorithm, in order to minimize the loss function, we chose the stochastic gradient descent (SGD) optimizer with a *momentum* of 0.9.

4.1.1 MobileNetV2 and Imagenet

Fine-tuning is a powerful method to obtain image classifiers on your own custom datasets from pretrained CNNs. So after exploring other ways to do the classification, we chose to use the ImageNet (Deng et al., 2009) weights in an already established architecture. In this study we try different architectures from the ones available in the `keras` package. Finally the network used is MobileNetV2 (Sandler et al., 2018).

Model	Top-1 Acc.	Top-5 Acc.	#Parameters	Depth
MobileNetV2	71.3%	90.1%	3.5M	105

TABLE 4.1: Stats of MobileNetV2.

The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input an MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer.

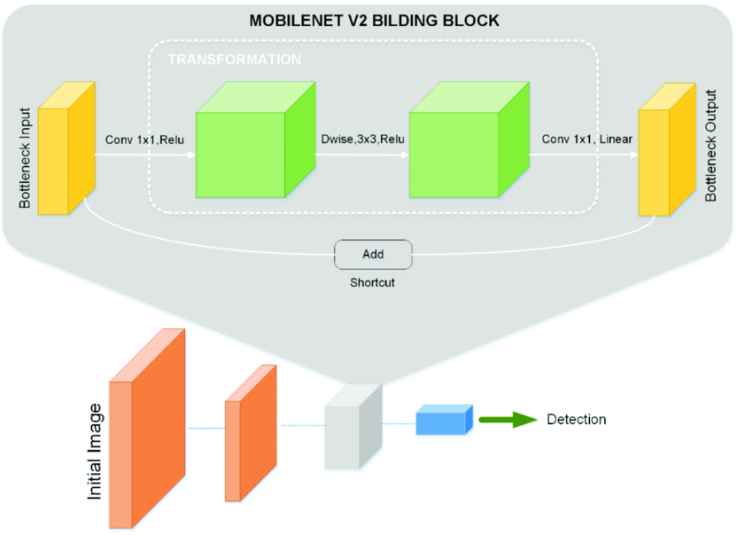


FIGURE 4.3: MobileNetV2 residual block.

The procedure to adapt this architecture to our problem is actually quite simple; just declare the dimension of the input as well as the output, and after loading the weights from the Imagenet database we will freeze the weights. This means that we will not change the values of the learned features in the convolutional layers (on MobileNetV2). Therefore, the dataset used should be as general as possible. The Imagenet weights meet this condition. After this we will couple the upper layers of the architecture adapted to our classifier which corresponds to the final layers of the network. This final classifyer consist in a first convolutional layer with 1×1 filters in order to reduce dimensionality in the filter dimension. Then we use the same architecture as in FabraCloudNet with a 5 neurons dense layer, using the softmax activation function. After the Flatten layer, we use a dropout of 50% to disable the half of the neurons, and avoid overfitting. For the final experiments, this mixed architecture has proven to be very effective.

4.2 Data Augmentation

Unfortunately, probably because it is a little explored problem in the field of deep learning, the volume of data we have to train our models is insufficient, even with the addition of the Fabra Clouds personal dataset, which consists of 2228 images. Therefore, in some experiments data augmentation techniques has been used in order to increase the number of different images, so that variability of the images in our dataset increases and, as a consequence, our models will be able to generalise and classify better the unseen images.
It should be noted that feature extraction, in this particular case is very sensitive to the environment and there will be restrictions on data augmentation transformations. That is, our models are intended to classify photographs taken from the ground which will always have a common general pattern; some ground or land will occupy part of the image while the vast majority will be sky with clouds present. Therefore, effects such as vertical flipping or very sharp rotation of angles can generate unwanted artefacts for the correct classification. Generally, during the experiments carried out, these three simple transformations have been applied with a certain degree of randomness:

- Horizontal flip,

FIGURE 4.4: **Horizontal Flip modification example.**

- Brightness range modification (from making the image a 15% darker to making it a 35% clearer),

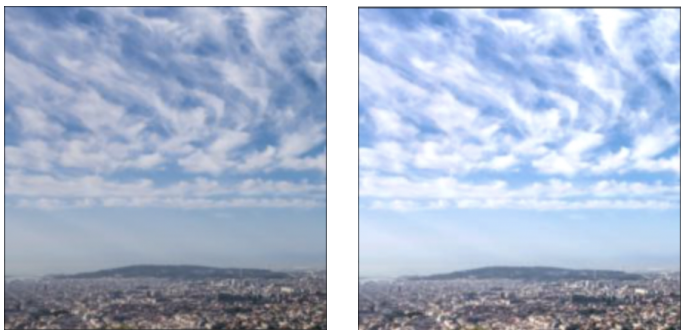
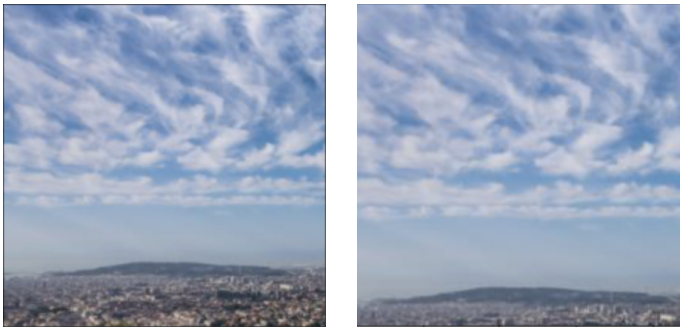


FIGURE 4.5: Bright modification example.

- Zoom In.



Original **Zoomed In**

FIGURE 4.6: "Zoom In" transformation example.

4.3 Vision Transformer (ViT)

In the paper *"Attention Is All You Need"* (Vaswani et al., 2017) an architecture called Transformer was introduced. As the title indicates, it uses the attention-mechanism. The attention-mechanism looks at an input sequence and decides at each step which other parts of the sequence are important. It sounds abstract, but let me clarify with an easy example: When reading this text, you always focus on the word you read but at the same time your mind still holds the important keywords of the text in memory in order to provide context.

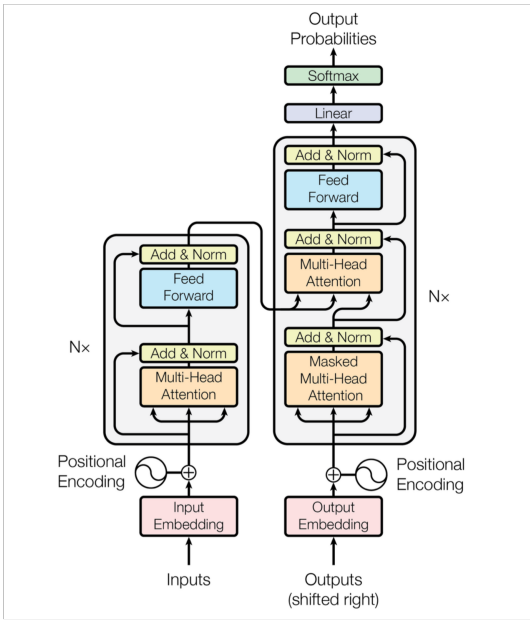


FIGURE 4.7: The transformer model architecture.

As this is a computer vision's oriented problem we will use a Vision transformer (ViT). A Vision Transformer is a transformer used in the field of computer vision that works based on the working nature of the transformers used in the field of natural language processing. The idea behind them is similar to the previous example

with the sentence. As in computer vision, we can use the patches of images as the token, we want to determine which of this parts are the most important to carry out the classification. Internally, the transformer learns by measuring the relationship between input token pairs. Generally speaking, a vision transformer performs the following steps to work.

- Firstly, split an image into patches.
- Then flatten the patches, producing lower-dimensional linear embeddings.
- Add positional embeddings.
- Feed the sequence as an input to a standard transformer encoder.
- Pretrain the model with image labels.
- Finetune on the downstream dataset for image classification.

The most notable drawback of vision tranformers is that they still require more data than convolutional neural networks. In the academic paper Dosovitskiy et al., 2020, the authors mention that Vision Transformers (ViT) are data-hungry. Therefore, pretraining a ViT on a large-sized dataset and fine-tuning it on medium-sized datasets (like ImageNet) is the only way to beat state-of-the-art Convolutional Neural Network models. The self-attention layer of ViT lacks locality inductive bias (the notion that image pixels are locally correlated and that their correlation maps are translation-invariant). This is the reason why ViTs need more data. On the other hand, CNNs look at images through spatial sliding windows, which helps them get better results with smaller datasets.

As is in our case, our dataset is not big enough to improve the expected results in the convolutional neural network. So we pretend to do the same as is stated in the following paper Lee, Lee, and Song, 2021, where the authors set out to tackle the problem of locality inductive bias in ViTs, by using the ideas of; Shifted Patch Tokenization and Locality Self Attention.

4.3.1 Shifted Patch Tokenization

In a ViT pipeline, the input images are divided into patches that are then linearly projected into tokens. Shifted patch tokenization (STP) is introduced to combat the low receptive field of Visual Transformers. The steps for Shifted Patch Tokenization are as follows:

- Start with an image.
- Shift the image in diagonal directions.
- Concat the diagonally shifted images with the original image.
- Extract patches of the concatenated images.
- Flatten the spatial dimension of all patches.
- Layer normalize the flattened patches and then project it.

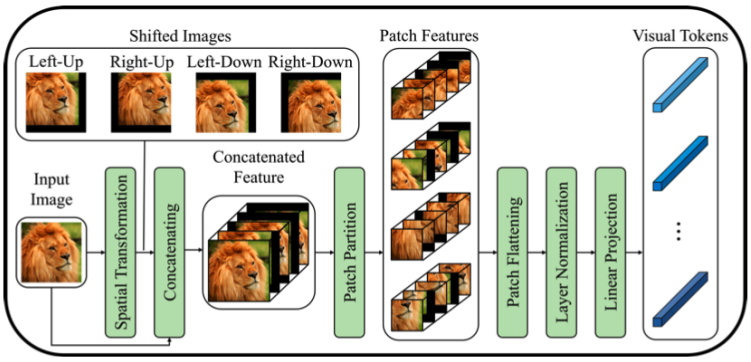


FIGURE 4.8: Shifted Patch Tokenization.

We can see how the actual tokenization is performed on our dataset as it is described in the following figure 4.9.

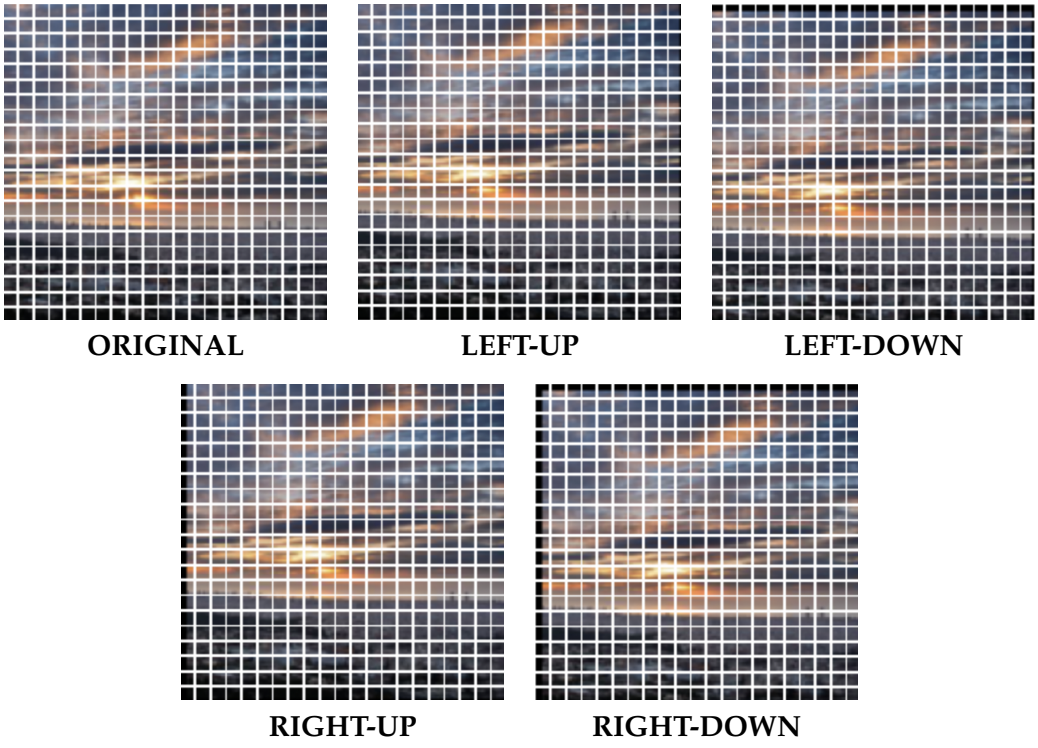


FIGURE 4.9: Shifted Patch Tokenization on actual data of Fabra Clouds dataset.

4.3.2 Locality Self Attention implementation

The attention equation is stated below.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The attention module takes three values; a query, a key, and a value. First, we compute the similarity between the query and key by doing the dot product. Then, the result is scaled by the square root of the key dimension. The scaling prevents the softmax function from having an overly small gradient. Softmax is then applied to

the scaled dot product to produce the attention weights. The value is then modulated via the attention weights. In self-attention, query, key and value come from the same input. The dot product would result in large self-token relations rather than inter-token relations. This also means that the softmax gives higher probabilities to self-token relations than the inter-token relations. To face this issue we have to mask the diagonal of the dot product. This way, we force the attention module to pay more attention to the inter-token relations. The scaling factor is a constant in the regular attention module. This acts like a temperature term that can modulate the softmax function. To use a learnable temperature term instead of a constant, is suggested by the autors.

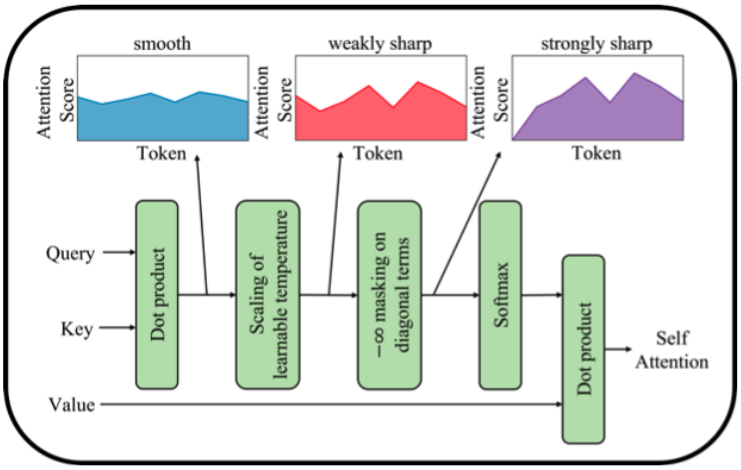


FIGURE 4.10: Locality Self Attention

4.4 Triplet Loss and Siamese Neural Networks

The notion behind is to understand how to represent data using distributed embeddings so that, in a high-dimensional vector space, contextually similar data points are projected close together while dissimilar data points are projected far apart. We can learn distributed embedding using the concepts of similarity and dissimilarity thanks to the triplet loss architecture (Hoffer and Ailon, 2014). This type of neural network architecture involves the training of many parallel networks that exchange weights. In order to construct distributed embeddings representation of input data, this input is routed through one network throughout prediction time.

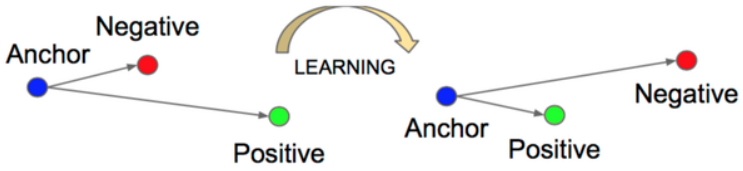


FIGURE 4.11: Similar images must be spatially closer than the dissimilar ones.

4.4.1 Triplet Loss

For Triplet Loss, the objective is to build triplets <anchor, positive, negative> consisting of an anchor image, a positive image (which is similar to the anchor image), and a negative image (which is dissimilar to the anchor image). The terms "similar" and "dissimilar" photographs can be defined in various ways. But this time, we will consider that, photos from the same class will be similar, while if the images are from a different class will be dissimilar.

So for example, in the dataset of FabraClouds we have ten different classes; Ac, Sc, Ns, Cu, Ci, Cc, Cb, As, Cs and St. The algorithm followed to generate the triplets is straight forward. First we need to select one sample of one class (Cb), and then pick another different but similar image (other sample from Cb). Note that this means to choose one different image from the same class. Now, images of the same classes are considered as similar, so one of them is used as an anchor and the other one as positive whereas images from the other class (e.g. Ac) is considered a negative image.

anchor **positive** **negative**

FIGURE 4.12: The triplets are constituted by the anchor, positive and negative images.

Thus, the loss function has the following form:

$$L(a, p, n) = max(0, D(a, p)^{-}D(a, n) + margin)$$

where the distance between the learned vector representations of x and y is denoted by $D(x, y)$. For example, $L2$ distance or $(1 - cosine\ similarity)$ can be used as a distance metric. The goal of this function is to maintain a smaller distance between the anchor and the positive than between the anchor and the negative (4.11).

4.4.2 Model architecture; three siamese CNN

The idea is to have 3 identical networks having the same neural net architecture and they should share weights. All networks should use the same weight vectors. D-dimensional vector representation can be learned using the D-number of neurons in the Deep Network's final FC layer (in our case is learning a representation of length 128). Anchor, Positive, and Negative pictures are sent across their respective networks, and common architecture is used to update weight vectors during back-propagation. During prediction time, any one of the networks is used to compute the vector representation of input data.

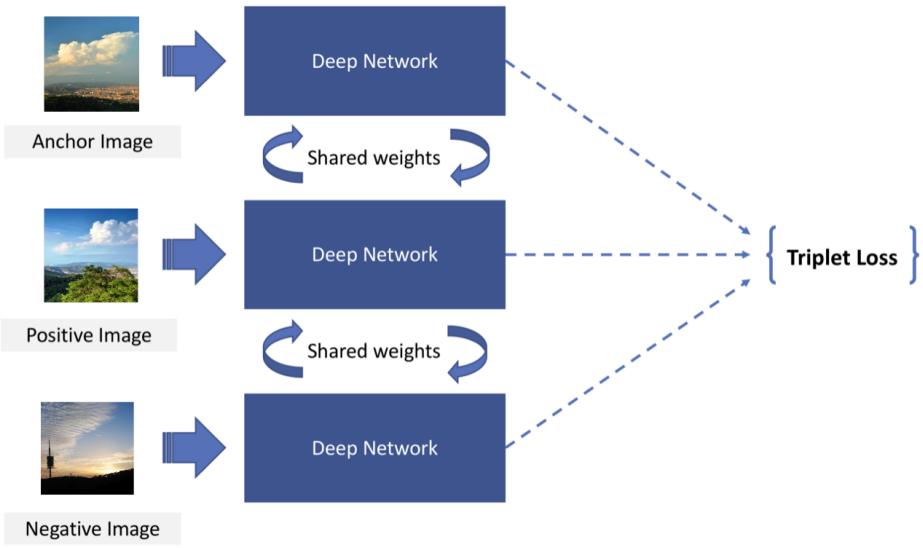


FIGURE 4.13: Triplet Loss architecture

The model not only learned to formulate clusters for different classes at the same time, but it's also successful in projecting similar-looking images into their neighborhood region. In the case of classification architecture, the model tries to learn a decision boundary between a pair of classes, but the model doesn't take care of the integrity between similar and dissimilar images within a class.

After training the triplet model, what has been done is to train a simple classifier from the embeddings of the train set. This classifier simply has the input layer of the same size as the compressed vectors (the embeddings have always been of size 128), and then a dense layer with as many units as there are different classes, with softmax activation function as usual.

4.5 Conformal Prediction

Classifying clouds presents a variety of problems. In first instance, even within the same genre there are different varieties of clouds. This differences between species of clouds of the same genre, makes it difficult for our supervised learning algorithms to extract features and classify them accurately, even so the features are extracted efficiently, there are different classes that are very similar in shape from the ground as can be seen in 6.1 in the section 6. In addition, it is possible that more than one genre or type of cloud may appear in an unseen image sample. For this reason, it has been decided to work within the framework of conformal prediction (Shafer and Vovk, 2007).

Conformal prediction is a technique for quantifying such uncertainties for artificial intelligence systems. In particular, given an input, conformal prediction estimates a prediction interval in regression problems and a set of classes in classification problems. Both the prediction interval and sets are guaranteed to cover the true value with high probability. Given an identically independently distributed (iid) dataset $((X_1, Y_1)...(X_n, Y_n))$ and a deep learning model represented as $f : X \rightarrow Y$ that has been trained, the goal here is to estimate prediction sets for model outputs.

To illustrate this idea, we will describe conformal prediction's steps and mathematical guarantees.

4.5.1 Identification of the score function; inconsistency quantification

Identify an appropriate score function $s(X, Y) \in R$ to measure the discrepancy between model outputs $\hat{y}$'s and labels y's. This score function is critical in that it actually decides what prediction sets we could get. This way, the resulting prediction sets whose values are within an L1-norm ball around the prediction $\hat{y}$; in classification problems, we could take $1 - \hat{y}_i$ as the score function, where $\hat{y}_i$ is the predicted logits for the true classic. This way, we will get a prediction set of classes whose predicted logits are greater than some given threshold.

4.5.2 Quantile of scores computation

Compute $\hat{e}$ as the $(1 - \alpha)$ quantile of the scores $\{s_1, \ldots, s_n\}$, where

$$s_1 = s(X_1, Y_1), ..., s_n = s(X_n, y_n)$$

In the full conformal prediction method, we need to train m models to compute the scores and construct prediction sets, where m is the number of possible values Y_{n+1} could take. This is undoubtedly computationally expensive. To lower the computation complexity, Inductive (split) Conformal Prediction could be used. Briefly, this method splits the whole training set into a proper training set and calibration set. Then, the model is only trained on the proper training set; and scores are solely computed on the calibration set. In this way, we only need to train the model once.

4.5.3 Prediction sets construction

Use the quantile to form the prediction sets for new examples,

$$\tau(X_{n+1}) = \{y : s(X_{n+1}, y) \leq \hat{e}\}$$

With the $(1 - \alpha)$ quantile $\hat{e}$, we could construct the prediction set for an input X_{n+1} by including values y's whose score with the input $s(X_{n+1}, y)$ is less or equal than $\hat{e}$.

4.5.4 Conformal prediction guarantees

Let Y_{n+1} be the true value. Y could be a class label in the classification problem. Let $\tau(X_{n+1})$ be a prediction set (or interval). We define that $\tau(X_{n+1})$ covers Y_{n+1} if Y_{n+1} is in $\tau(X_{n+1})$, i.e.,

$$Y_{n+1} \in \tau(X_{n+1})$$

Then, given a set of i.i.d samples

$$\{(X_1, Y_1), (X_2, Y_2,), \ldots, (X_n, Y_n)\},$$

the conformal prediction set satisfies the following coverage guarantee, i.e.,

$$P(Y_{n+1} \in \tau(X_{n+1})) \geq 1 - \alpha$$

A proof of the coverage guarantee based on exchangeability i.i.d assumption could be found in the paper Angelopoulos and Bates, 2021. Note that, in this proof, the $(1 - \alpha)$ level is changed to $(n + 1)(1 - \alpha)/n$ to account for the finite sample case.

Chapter 5

Experimentation and discussion of the results

This chapter presents some of the experiments carried out and the results obtained with different datasets. In other words, we are going to see how the models behave around different ways of classifying clouds.

The datasets that have been used can be divided into two groups. Those that purely seek to classify from among the genera stipulated by the World Meteorological Organisation are mainly the Cirrus Cumulus Nimbus Stratus (abbreviated with the acronym CCSN) and FabraClouds datasets. They present the following categories; Ac, Sc, Ns, Cu, Ci, Cc, Cb, As, Ct, Cs, St. As we can see in the figure 3.3, in FabraClouds there is a lack of samples of two of the ten classes we want to identify; Cirrostratus (Cs) and Nimbostratus (Ns). So these two classes has been removed from this dataset, and will be performing experiments with the remaining classes. The Cirrostratus and Nimbostratus class will be also omitted for CCSN dataset. On the other hand, we have those which distinguish from simpler visual patterns. These datasets summarise the simplest ways of classifying cloud morphological characteristics; they are the Swimcat and FabraSwimcat datasets. As it is stated before, this last way of classification may be insufficient to fulfil the usefulness of cloud identification, but it is much easier for models to learn this simpler set of patterns. As already mentioned, there are some drawbacks if we want to classify clouds according to genus due to the similarities between some of them, and other problems such as the presence of two types in the same picture (among others). Therefore, for the latter case we will see that the conformal prediction framework makes sense in the case of the classification of the 10 (in some experiments we consider only 8) cloud genera. Without further ado, let us review the results.

Although the models have been training with all available data, I will put emphasis on the classifications for the data belonging to the Observatori Fabra, as this is where the models are supposed to work.

For all trainings, all three RGB channels have been taken into account. The reasoning is that the genesis or appearance of some cloud types can be related to the temperature or conditions that are reached at a certain time of the day. As we know, due to the tilt of the sun in certain time zones, the colours of the sky can vary. For example, it is very common at dawn or dusk to have images with a more orange-coloured sky.

5.1 Cloud genera classification; CCSN and FabraClouds

For both types of classification, the models described in section 4 have been used and tested, but this time the model that is giving the best performance is the MobileNetV2 pretrained with Imagenet weights with a slight modification. Firstly we have trained the network with CCSN, with the mentioned model. It is intended to train this network for about 200 epochs, with a fairly low batch size of a value of 8 and a learning rate of 1e−3. Any scheduler for the learning rate has been used, as after many experimentation using a step decay as well as an exponential decay, the results seem to be worse. In addition, an early stopping has been configured with a value for patience equal to 20 epochs. This means that if the results on the validation set do not improve after 20 iterations we will stop the training. Finally results from that training are the ones shown in the figure 5.1.

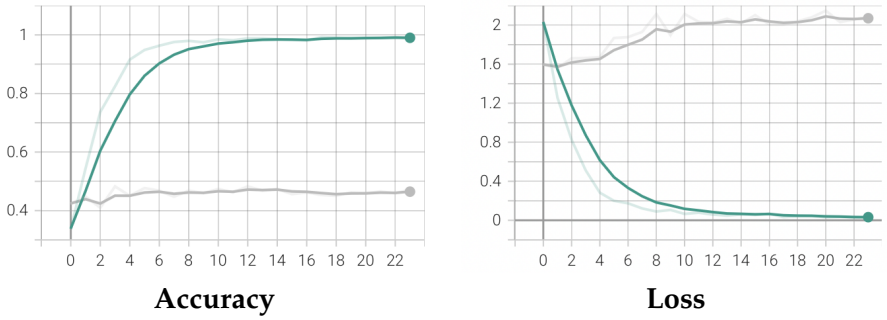


FIGURE 5.1: Accuracy and error loss during the first training with the dataset CCSN. The model was fine tuned with ImageNet weights.

We can see that there is clearly a very early overfitting in this case as the validation loss is increasing immediately. In consequence the accuracy of the model do not improve at all. But, by modifying the values of the hyperparameters, the results with this dataset do not improve. Due to the early stopping the training stops after 22 epochs have passed. Now, we will perform a similar experiment but this time with FabraClouds dataset. In the previous experiment the layers of MobileNetV2 were frozen, but now I wanted to try something different and set all the layers as trainable. So now the upper layers of the MobileNetV2 network can learn about the input data but initialized with the ImageNet weights 5.2.

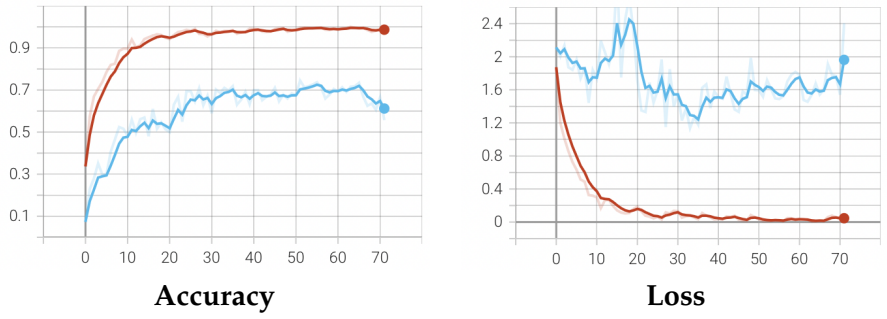


FIGURE 5.2: Accuracy and error loss during the second training. This time we have trained the CNN with the MobileNetV2 initialized with the ImageNet weights.

We now test the model against the test set, and the result of the confusion matrix is as shown in the figure 5.3 below.

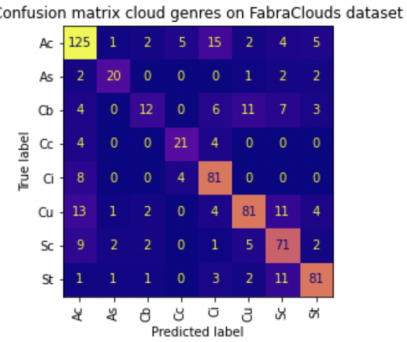


FIGURE 5.3: Confusion matrix on the classification of the 8 genres on FabraClouds dataset. These genres are; Ac, As, Cb, Cc, Ci, Cu, Sc, St.

	precision	**recall**	**f1-score**	**support**
Ac	0.75	0.79	0.77	159
As	0.80	0.74	0.77	27
Cb	0.63	0.28	0.39	43
Cc	0.70	0.72	0.71	29
Ci	0.71	0.87	0.78	93
Cu	0.79	0.70	0.74	116
Sc	0.67	0.77	0.72	92
St	0.84	0.81	0.82	100

TABLE 5.1: Key performance metrics of the best model by categories.

This is the best result that has been obtained after many experimentation with all the explained models; the pretrained MobileNetV2 model is giving a global accuracy of the 75%. For these experiments, the data augmentation technique did not seem to help improve the predictions at all. In order not to stress the memory of the computer too much, it has been decided to keep the number of images of the original dataset.

From the algorithms implemented to perform the conformal prediction, with 95% confidence, the model tells us that for an image of the Stratocumulus category the predicted classes and the confidence percentages are the ones shown in 5.4.

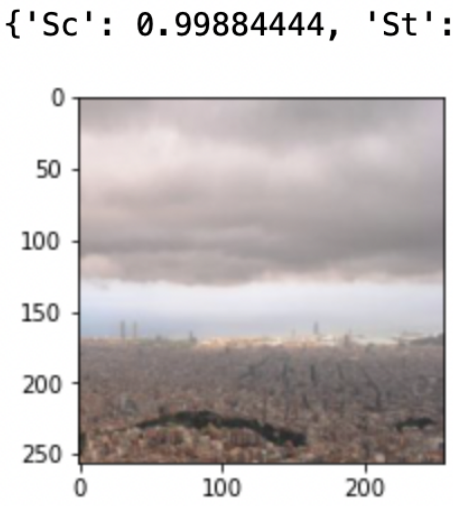


FIGURE 5.4: This is the output for one sample of the test set. The image belongs to the class Stratocumulus. The model is saying that with almost 99% of confidence is from the correct class, but it doubts a little with the class Stratus (St)

As we can see in the confusion matrix, several examples of the Stratus class are incorrectly detected as Stratocumulus. Even so, such a doubt seems to make sense since they are classes whose morphology (in most of the cases but not all) is very similar.

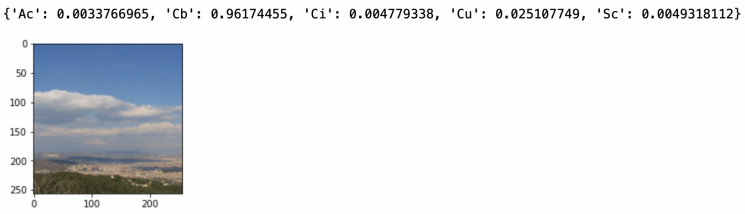
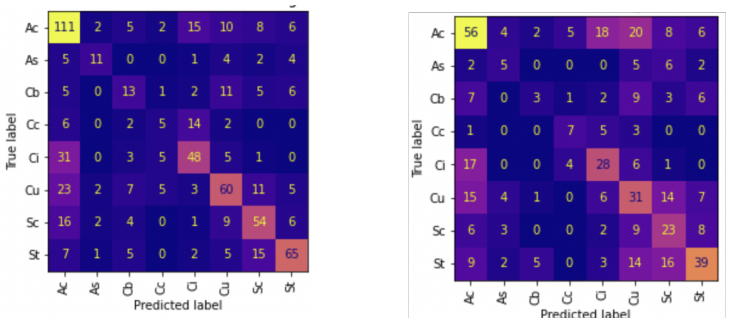


FIGURE 5.5: This is the output for one sample of the test set. The image belongs to the class Stratocumulus as in the previous example. The model is saying that is almost sure that the class predicted is cumulonimbus (Cb). But this time fails. However, we can see that Sc appears in the set of possible solutions in the third position.

Here we can also see that when the model is more uncertain (the set returned as a solution has a higher dimension) it tends to give a wrong answer in the end. Moreover, from the picture at first glance it seems reasonable that it would be wrong given the similarity between the examples of the returned possible classes.

For other models such as the vision tranformer or the triplet model, this type of genre classification results in a worst classification as we can see in the figure 5.6 below.



ViT **Triplet Loss**

FIGURE 5.6: Output confusion matrix of the Vision Transformer and the Triplet Loss model.

5.2 Swimcat categories classification; Swimcat and FabraSwimcat

For the latter type of classification in five categories (Clear Sky, Patterned Clouds, White Thick Clouds, Dark Thick Clouds and Veil Clouds) the results are much better. This is because, as mentioned above, these features are much more visual and more detectable by the vision models. That is, there is a greater difference or distance between the classes. For the swimcat dataset it seems that all models perform very well. I personally find the representation that the triplet model achieves interesting. Before the training of the triplet model the training examples are distributed on the plane like below.

Swimcat embeddings before training **FabraSwimcat embeddings before training**

FIGURE 5.7: Compact representation in the plane of the training images of Swimcat and FabraSwimcat dataset.

Then, after the generation of the triplets as indicated in section 4, and the training of the model, the representation that we obtain is the following one.

Swimcat embeddings after training **FabraSwimcat embeddings after training**

FIGURE 5.8: More compact representation in the plane of the training images of Swimcat dataset and FabraSwimcat, after training the model by 16000 triplets.

This is very interesting, because in this way we can have a representation of the similarity between the images in the dataset. That is, this time we can measure exactly how similar one class is to another.

After this appreciation, although the triplet model and the ViT model perform well, the convolutional neural network MobileNetV2 pretrained with ImageNet, still gives better results also on this occasion as we can see in the following figure 5.9.

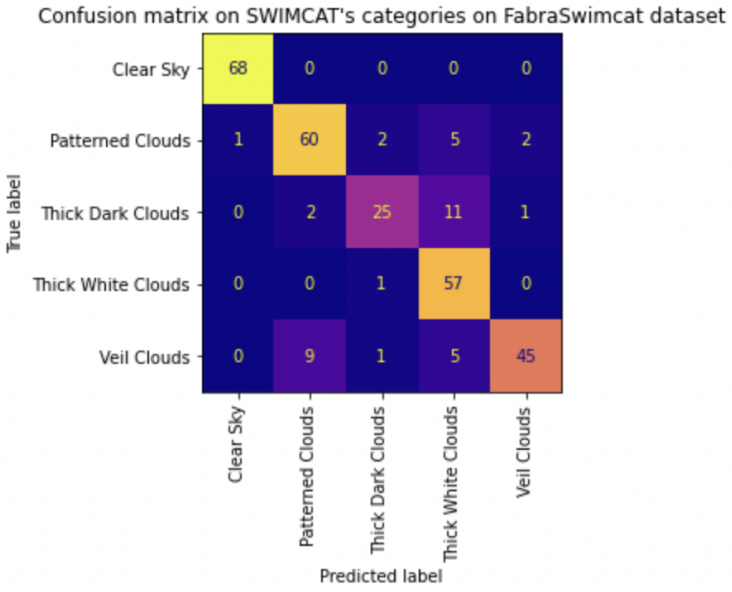
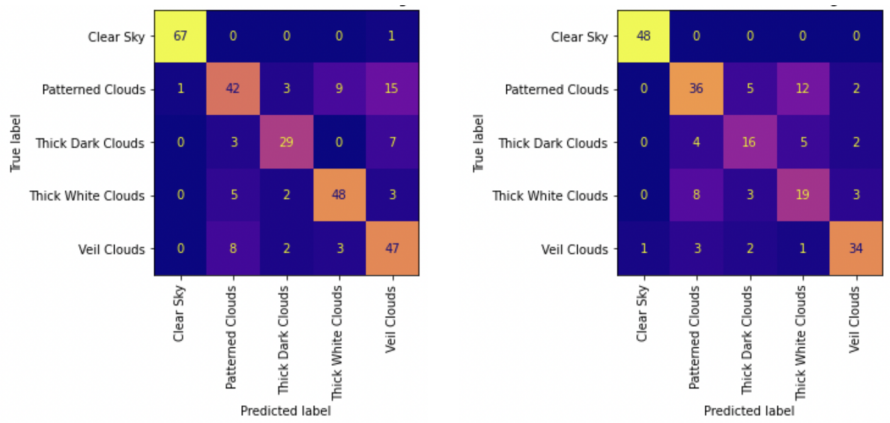


FIGURE 5.9: Confusion matrix on the classification of the 5 classes on FabraSwimcat dataset, by doing fine tunning with MobileNetV2 and Imagenet.

	precision	**recall**	**f1-score**	**support**
Clear Sky	0.99	1.00	0.99	68
Patterned Clouds	0.85	0.86	0.85	70
Thick Dark Clouds	0.86	0.64	0.74	39
Thick White Clouds	0.73	0.98	0.84	58
Veil Clouds	0.94	0.75	0.83	60

TABLE 5.2: Key performance metrics of the best model by categories (2).



ViT **Triplet Loss**

FIGURE 5.10: Output confusion matrix of the Vision Transformer and the Triplet Loss model for FabraSwimcat test set.

Thus we can see that the winning models for the particular problem of cloud identification, are not even the triplet model neither the vision transformer; the best model is obtained through a convolutional neural networks. As mentioned above, the set of experiments carried out has been extensive and exhaustive, aiming to better classify the images belonging to the fabra observatory.

Chapter 6

Conclusions

In this chapter we will make some important observations that have determined the direction and outcome of this project, as well as to give indications of possible ways for the continuation of this study.

In first instance, we have been able to come up with a good (but also improvable) classifier consisting mainly of the MobileNetV2 residual network architecture pretrained with the ImageNet weights, specialised in distinguishing the following cloud genera; Ac, Sc, Cu, Ci, Cc, Cb, As, Ct, St. Also, the CNN along the other models trained (ViT and Triplet Loss model) are performing well in a Swimcat-like category classification.

After all the work done we can state that classifying clouds presents a variety of problems. On the one hand, no two clouds are alike (even within the same genre there are different varieties), which makes it difficult for supervised learning algorithms to extract features and classify them accurately, even so the features are extracted efficiently, there are different classes (if we talk on terms of ten/eight genre classification) that are very similar in shape from the ground (as can be seen in 6.1 between the Altocumulus and Cirrocumulus classes). In addition to this, in the same image taken from the ground, there may be a cloud in an intermediate state between two of the two types of clouds to be classified, or two types of cloud may coexist in the same image (ideally if the dataset is well labelled by an expert this should not happen). Therefore, to cover such problems, it has been decided to apply a mechanism to measure the uncertainty of the trained models (Balasubramanian, Ho, and Vovk, 2014), giving a more flexible response to the classification, as we are giving a set of classes where the solution to the problem is located. In this way we will also be able to understand what the model has been able to learn through its training.

But without a doubt, the biggest problem we face in classifying clouds is that there is very little published information on cloud labelled images taken from the ground. The number of images available to us today is not sufficient, so we cannot train a model that generalises and classifies cloud genera well. On the other hand, if we want to classify simpler patterns, the current information is sufficient.

(Ac) **(Cc)**

FIGURE 6.1: **(Ac)** Altocumulus vs. **(Cc)** Cirrocumulus

6.1 Future work

Climate change is a major problem affecting the entire globe. The global rise in temperature has disastrous consequences that threaten the survival of the Earth's flora and fauna, including the human specie. Among the impacts of climate change are the melting of the ice mass at the poles that at the same time causes the rise of the sea level, resulting in flooding and threatening coastal areas such as the city of Barcelona. In addition, this alteration of temperatures also increases the occurrence of more violent weather events, droughts, fires, the death of animal and plant species, the overflowing of rivers and lakes, the emergence of climate refugees, and the destruction of livelihoods and economic resources. Clouds are a clear indicator of the climate of a region, so their proper identification is another tool to help us understand changes in the environment over time. In a complementary way, deep learning tools such as the ones we have presented can be very useful to achieve this goal, as we have seen that the main problem is the lack of available data.

Much of the effort from now on should be devoted to getting more images properly tagged. In this way, the data for training deep learning models could be enough to achieve our goal, despite the problems presented by the classification of clouds in particular. Initiatives such as citizen science, which involves citizens in data collection, can help us to expand the cloud image database. We can even develop software that helps us to keep track of the data that is collected and with the help of professionals we can label it appropriately. One idea that could be implemented is the development of an application to simply request access to the camera of our smart phone to, at a given moment, point it at the sky and take pictures of it. Lately, we can then use some system to verify the correctness of the data. In fact, there is already an app that do this kind of stuff; CloudSpotter.

From the automatic classification of clouds, new sources of study can be discerned. For example, the study of cloud evolution over a given period of time by means of a sequential data set, such as photos taken at a relatively high frequency, or videos. In this way we can acquire more predictive power through the detection of other weather patterns that may occur from cloud analysis.

Appendix A

Cloud traditional classification algorithms

A.1 Depending to their height

A.1.1 Lower Clouds (C_L)

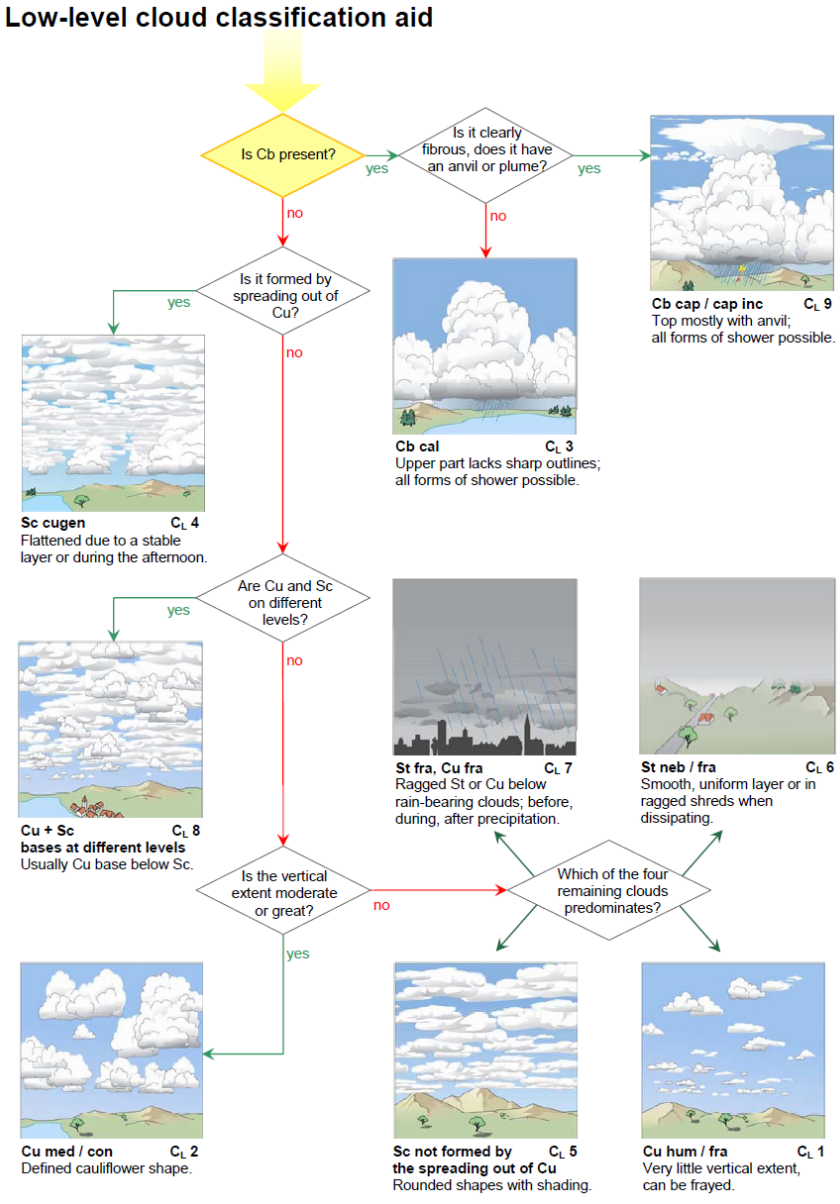


FIGURE A.1: Flowchart to classify low clouds.

A.1.2 Medium Clouds (C_M)

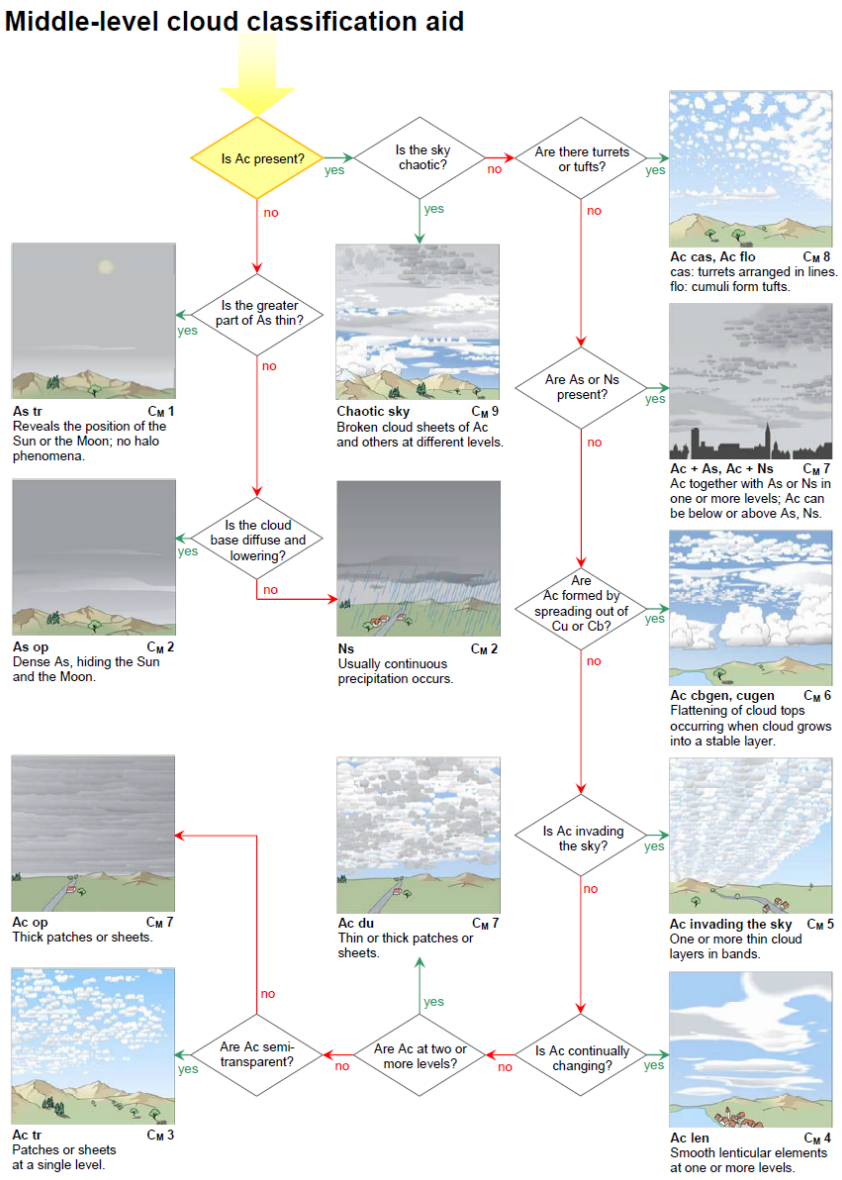


FIGURE A.2: Flowchart to classify medium clouds.

A.1.3 High Clouds (C_H)

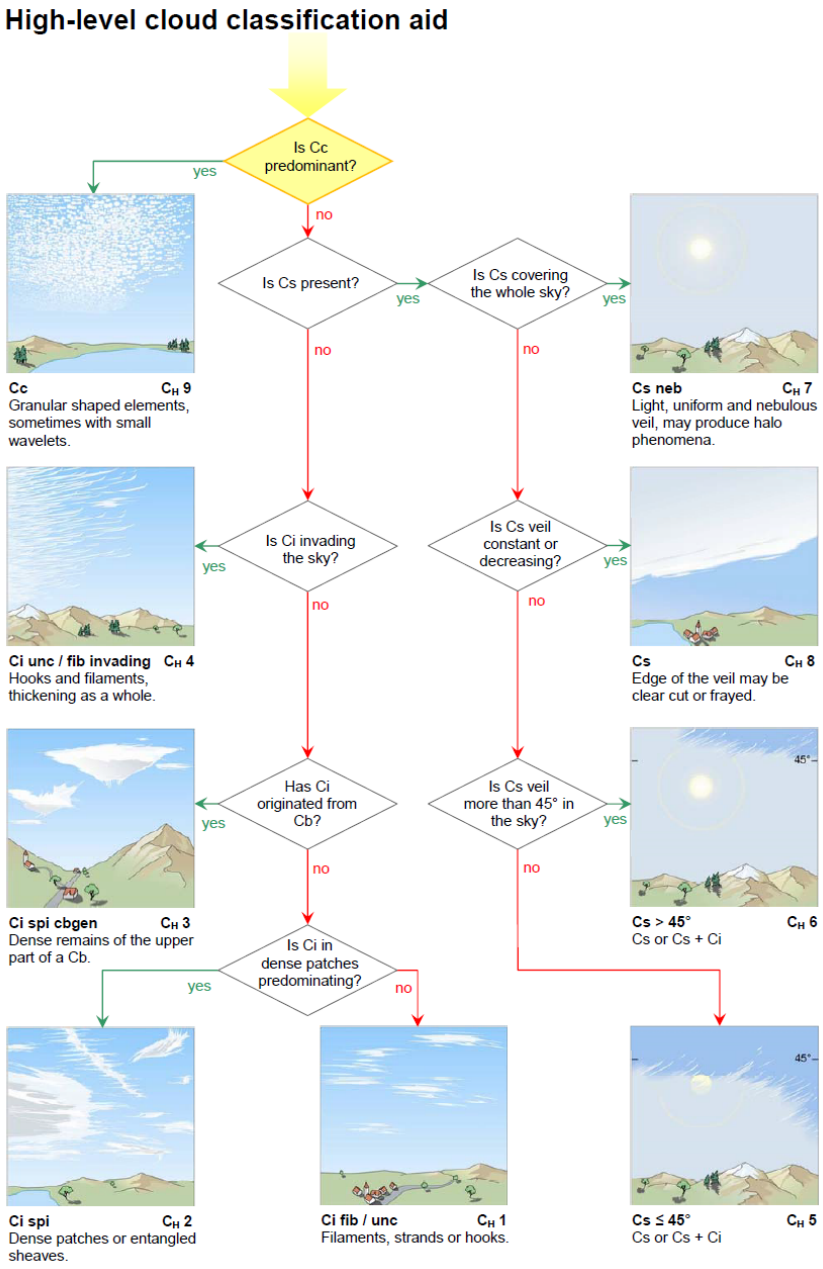


FIGURE A.3: Flowchart to classify high clouds.

Bibliography

Angelopoulos, Anastasios N. and Stephen Bates (2021). *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. DOI: 10.48550/ARXIV.2107.07511. URL: https://arxiv.org/abs/2107.07511.

Balasubramanian, Vineeth, Shen-Shyang Ho, and Vladimir Vovk (2014). *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes.

Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Dev, Soumyabrata, Yee Hui Lee, and Stefan Winkler (2015). "Categorization of cloud image patches using an improved texton-based approach". In: *2015 IEEE international conference on image processing (ICIP)*. IEEE, pp. 422–426. URL: https://stefan.winkler.site/Publications/icip2015cat.pdf.

Dosovitskiy, Alexey et al. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929.

Hoffer, Elad and Nir Ailon (2014). *Deep metric learning using Triplet network*. DOI: 10.48550/ARXIV.1412.6622. URL: https://arxiv.org/abs/1412.6622.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Lee, Seung Hoon, Seunghyun Lee, and Byung Cheol Song (2021). *Vision Transformer for Small-Size Datasets*. DOI: 10.48550/ARXIV.2112.13492. URL: https://arxiv.org/abs/2112.13492.

Martin-Brualla, Ricardo et al. (2020). *NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections*. DOI: 10.48550/ARXIV.2008.02268. URL: https://arxiv.org/abs/2008.02268.

Radford, Alec et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. DOI: 10.48550/ARXIV.2103.00020. URL: https://arxiv.org/abs/2103.00020.

Sandler, Mark et al. (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: DOI: 10.48550/ARXIV.1801.04381. URL: https://arxiv.org/abs/1801.04381.

Shafer, Glenn and Vladimir Vovk (2007). "A tutorial on conformal prediction". In: DOI: 10.48550/ARXIV.0706.3188. URL: https://arxiv.org/abs/0706.3188.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.