# Day 1 – Sessions 1 & 2: Sentinel Data Deep Dive & AI/ML Core Concepts

## Session 1: Copernicus Sentinel Data Deep Dive & Philippine EO Ecosystem

### Module 1: Copernicus Program Overview; Sentinel-1 & Sentinel-2

The **Copernicus Programme** is the European Union’s flagship Earth observation initiative, providing free and open access to environmental satellite data[[1]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=). Central to Copernicus are the **Sentinel satellites**, designed to systematically monitor land, ocean, and atmosphere. This module focuses on **Sentinel-1** (radar) and **Sentinel-2** (optical) – two missions particularly relevant to Earth Observation (EO) analysts.

**Sentinel-1** is a constellation of polar-orbiting satellites carrying a C-band Synthetic Aperture Radar (SAR) instrument[[2]](https://dataspace.copernicus.eu/data-collections/sentinel-data/sentinel-1#:~:text=The%20Sentinel,images%20in%20all%20weather%20conditions). SAR actively emits microwave signals and measures the backscatter, enabling imaging **day or night and through clouds or rain**, which is crucial for all-weather monitoring[[2]](https://dataspace.copernicus.eu/data-collections/sentinel-data/sentinel-1#:~:text=The%20Sentinel,images%20in%20all%20weather%20conditions). Sentinel-1 operates in multiple imaging modes with varying swath widths and resolutions: for example, **Interferometric Wide Swath (IW)** mode covers ~250 km with ~5 × 20 m spatial resolution, while **Extra-Wide (EW)** swath covers ~400 km at ~25 × 100 m[[3]](https://www.eoportal.org/satellite-missions/copernicus-sentinel-1#:~:text=SM%20mode%20has%20an%2080,m%20x%20100%20m). The mission provides **dual polarization** data (typically VV/VH or HH/HV polarizations) and has a **short revisit cycle** of 6 days (with two satellites; ~12 days at the equator)[[4]](https://www.pna.gov.ph/articles/1136226#:~:text=Philippine%20Sky%20Artificial%20Intelligence%20Program,widespread%20processing%20systematically%20and%20effectively)[[5]](https://www.sciencedirect.com/science/article/pii/S0034425722003169#:~:text=Effect%20of%20the%20temporal%20baseline,Potin%20et). This frequent revisit and SAR’s cloud-penetrating ability make Sentinel-1 ideal for applications like **flood mapping, forest biomass estimation, and land deformation monitoring**. Sentinel-1 data products include **Level-1 GRD (Ground Range Detected)** images – multi-look, amplitude-only data with radiometric and geometric corrections applied – and **Level-1 SLC (Single Look Complex)** data containing complex pixels (preserving phase information for interferometry). These standard products enable a wide range of analyses, from basic viewing to advanced interferometric SAR (InSAR) processing.

**Sentinel-2** is a pair of polar-orbiting satellites that carry a **Multispectral Instrument (MSI)** for optical imaging[[6]](https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi#:~:text=Sentinel,bands%20at%2060%20meter%20resolution). Sentinel-2 provides high-resolution **multispectral imagery in 13 spectral bands**, spanning the visible, near-infrared (NIR), and shortwave infrared (SWIR) ranges[[7]](https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/#:~:text=Dedicated%20to%20supplying%20data%20for,B09%2C%20B10%2C%20B11%20and%20B12)[[8]](https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/#:~:text=versatile%20set%20of%2013%20spectral,B09%2C%20B10%2C%20B11%20and%20B12). Notably, it captures four visible/NIR bands at **10 m** resolution, six red-edge/SWIR bands at **20 m**, and three atmospheric bands (for aerosol and water vapor) at **60 m**[[9][10]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=,meter%20pixel%20size). The **spatial resolution** thus varies by band (10/20/60 m), and the **swath width** is 290 km, allowing broad area coverage[[7]](https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/#:~:text=Dedicated%20to%20supplying%20data%20for,B09%2C%20B10%2C%20B11%20and%20B12). Sentinel-2’s **revisit frequency** is 5 days globally with two satellites (10 days per satellite)[[11]](https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi#:~:text=13%20bands%3A%20442.3%20nm%20,4%20nm%20%28see%20table), enabling near-biweekly observations of any location (and more frequent in mid-latitudes due to overlapping swaths). The combination of **fine spatial detail, multi-band spectral information, and frequent revisits** makes Sentinel-2 a workhorse for tasks like **land cover classification, vegetation health monitoring, water quality assessment, and disaster mapping**. For instance, it distinguishes vegetation vigor using the red-edge bands and NIR (important for indices like NDVI), while SWIR bands aid in detecting moisture and burn scars.

[[12]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=Figure%202%20Sentinel,Atmospheric)[[13]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=correction%20was%20conducted%20using%20Sen2Cor%2C,values%2C%20band%20reflectance%20in%20the)*Sentinel-2’s multispectral bands are strategically positioned across the electromagnetic spectrum.* **Figure 1** shows the Sentinel-2 band layout, illustrating how bands cover specific wavelength ranges to capture distinct information (e.g. Band 4 – red, Band 8 – NIR, Band 12 – SWIR) and their respective spatial resolutions. Each band’s design serves a purpose: for example, the red-edge bands (5, 6, 7, 8A) target the vegetation red-edge region for plant health studies, while Band 10 is a **cirrus cloud detection** band at 1.37 µm[[10]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=,meter%20pixel%20size)[[14]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=B8a%2020%20m%20865%20nm,SWIR). By providing consistent, calibrated imagery, Sentinel-2 enables comparison over time and across regions, which is vital for change detection and long-term environmental monitoring.

[[15]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=The%20following%20two%20main%20products,26)[[16]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=%2A%20Level,processor%20from%20ESA%27s%20SNAP%20Toolbox)Beyond raw imagery, Sentinel missions offer *standard data products*. Sentinel-2 imagery is distributed as **Level-1C** (Top-of-Atmosphere reflectance, orthorectified to UTM 100 km grid tiles) and **Level-2A** (Bottom-of-Atmosphere surface reflectance after atmospheric correction). Level-1C is approximately 500 MB per 100×100 km tile and is radiometrically and geometrically corrected, ready for further processing[[15]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=The%20following%20two%20main%20products,26). Level-2A is considered an **Analysis Ready Data (ARD)** product suitable for direct use in downstream applications[[17]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=product%20can%20be%20obtained%20from,by%20the%20user%20with%20the); it can be obtained from the Copernicus data platform or generated via the Sen2Cor processor[[18]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=instructions%20dataspace.copernicus.eu%20.%20%2A%20Level,processor%20from%20ESA%27s%20SNAP%20Toolbox). Sentinel-1 data products are processed to **Level-1 GRD** (detected, multi-looked intensity images) or **Level-1 SLC** (complex images preserving phase). Users can choose based on application: for example, flood mapping might use GRD intensity images, whereas ground deformation or terrain mapping would require SLC for interferometric analysis.

Accessing Copernicus data is made easy through multiple platforms. The official **Copernicus Open Access Hub** (now succeeded by the Data Space Ecosystem) provides free downloads of Sentinel-1 and Sentinel-2 data for any user. Additionally, cloud-based platforms have made data access even more convenient. For instance, **Google Earth Engine (GEE)** hosts Sentinel-1 GRD and Sentinel-2 Level-1C collections in its public catalog, allowing users to query and process imagery on the fly without manual download. This means an analyst can, within minutes, obtain a Sentinel-2 image composite for a given date range over the Philippines and apply a land classification algorithm in the cloud. Other platforms like Amazon Web Services (AWS) and Sentinel Hub also mirror Sentinel data. The key point is that **Copernicus data are open and accessible**, enabling broad use – from local agencies to international researchers[[1]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=). Later in this session, we will highlight the Philippines’ own efforts in locally mirroring and distributing Copernicus data (the **CopPhil Mirror Site**).

In summary, the Copernicus Sentinel missions provide a **comprehensive EO toolkit**: Sentinel-1’s all-weather radar sees through clouds and darkness for dynamic phenomena (floods, forest cover change, ground movement), while Sentinel-2’s optical sensor offers rich spectral detail for classifying and assessing Earth’s surfaces. Understanding their characteristics – spatial resolution, spectral bands, revisit frequency – and the available product levels is fundamental for EO professionals to select the right data and preprocessing needed for their AI/ML tasks.

**Figure 1 – Sentinel-2 Spectral Bands and Resolutions:** *Sentinel-2’s MSI covers 13 spectral bands from visible light through shortwave infrared. Four bands (blue, green, red, NIR) are 10 m resolution; six bands (vegetation red-edge and two SWIR) are 20 m; and three bands (coastal aerosol, water vapor, cirrus) are 60 m. This layout enables applications ranging from natural color imaging to vegetation and moisture indices.*[[12]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=Figure%202%20Sentinel,Atmospheric)[[13]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=correction%20was%20conducted%20using%20Sen2Cor%2C,values%2C%20band%20reflectance%20in%20the)

*Figure 1: Sentinel‑2 spectral band layout (wavelengths in µm). Each bar represents a spectral band (B1–B12) and its bandwidth, color-coded by resolution (10 m bands in green, 20 m in orange, 60 m in red). The bands span visible (VIS), near-infrared (NIR), and shortwave-infrared (SWIR) regions, providing a versatile dataset for Earth observation.*

### Module 2: The Philippine EO Landscape

Earth observation in the **Philippines** is supported by a growing ecosystem of agencies and platforms that complement the Copernicus data stream. This module introduces key Philippine organizations involved in EO and how their local datasets (e.g. hazard maps, land cover data, weather information) can enhance AI/ML projects when combined with Sentinel imagery.

* **PhilSA (Philippine Space Agency):** Established in 2019, PhilSA is the national space agency responsible for centralizing space policy and programs. It operates or partners in various EO initiatives. One notable platform is the **Space Data Dashboard (Space+ SDD)** – an online portal that provides access to satellite-derived information for the public and institutions[[19]](https://www.facebook.com/PhilSpaceAgency/posts/-%F0%9D%97%9C%F0%9D%97%96%F0%9D%97%AC%F0%9D%97%A0%F0%9D%97%9C-%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%81%F0%9D%97%AE-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%80%F0%9D%97%B5%F0%9D%97%AF%F0%9D%97%BC%F0%9D%97%AE%F0%9D%97%BF%F0%9D%97%B1-%F0%9D%97%B9%F0%9D%97%AE%F0%9D%98%82%F0%9D%97%BB%F0%9D%97%B0%F0%9D%97%B5-%F0%9D%97%B1%F0%9D%98%82%F0%9D%97%BF%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B4-%F0%9D%98%81%F0%9D%97%B5%F0%9D%97%B2-%F0%9D%97%A3%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%B9%F0%9D%97%B6%F0%9D%97%BD%F0%9D%97%BD%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B2%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2%F0%9D%97%AA%F0%9D%97%B2%F0%9D%97%B2%F0%9D%97%B8-%F0%9D%97%98%F0%9D%98%85%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%AF%F0%9D%97%B6%F0%9D%98%81the-phi/925350012971936/#:~:text=Philippine%20Space%20Agency%20,2024%20at%20the%20Quantum%20Skyview). Launched in 2024, the Space Data Dashboard aims to *“democratize space data”* by offering user-friendly visualization of EO data layers (e.g. land cover maps, disaster footprints)[[19]](https://www.facebook.com/PhilSpaceAgency/posts/-%F0%9D%97%9C%F0%9D%97%96%F0%9D%97%AC%F0%9D%97%A0%F0%9D%97%9C-%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%81%F0%9D%97%AE-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%80%F0%9D%97%B5%F0%9D%97%AF%F0%9D%97%BC%F0%9D%97%AE%F0%9D%97%BF%F0%9D%97%B1-%F0%9D%97%B9%F0%9D%97%AE%F0%9D%98%82%F0%9D%97%BB%F0%9D%97%B0%F0%9D%97%B5-%F0%9D%97%B1%F0%9D%98%82%F0%9D%97%BF%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B4-%F0%9D%98%81%F0%9D%97%B5%F0%9D%97%B2-%F0%9D%97%A3%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%B9%F0%9D%97%B6%F0%9D%97%BD%F0%9D%97%BD%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B2%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2%F0%9D%97%AA%F0%9D%97%B2%F0%9D%97%B2%F0%9D%97%B8-%F0%9D%97%98%F0%9D%98%85%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%AF%F0%9D%97%B6%F0%9D%98%81the-phi/925350012971936/#:~:text=Philippine%20Space%20Agency%20,2024%20at%20the%20Quantum%20Skyview). PhilSA also inherited the **PEDRO Center** (Philippine Earth Data Resource Observation Center), a ground station network that downloads imagery from satellites (including Diwata microsatellites and others). Through PEDRO and data-sharing agreements, PhilSA helps ensure local availability of imagery from various sources. In summary, PhilSA provides the national infrastructure for space data and works with agencies like ESA to augment local capacity – as seen in the Copernicus Philippine (CopPhil) cooperation discussed shortly.
* **NAMRIA (National Mapping and Resource Information Authority):** NAMRIA is the government’s central mapping agency. It produces base maps and thematic maps of the Philippines – including topographic maps, administrative boundaries, and crucially **national land cover maps and hazard maps**. For example, NAMRIA publishes a **Land Cover Dataset** (with classes like forest, agriculture, built-up, water) that is updated periodically and serves as ground truth or training data for AI models. It also collaborates on hazard mapping (flood and storm surge hazard maps) which incorporate historical EO data and surveys. These products are accessible through the **Philippines Geoportal**, an online platform NAMRIA manages to disseminate geospatial data. For an EO professional, NAMRIA’s data can significantly enrich Sentinel-based analyses: e.g., one could use the NAMRIA land cover map to label training samples for a Sentinel-2 classification model, or overlay NAMRIA flood hazard zones with Sentinel-1 flood extent results for validation.
* **DOST-ASTI (Department of Science and Technology – Advanced Science and Technology Institute):** DOST-ASTI is a research institute that leads many of the country’s EO and AI initiatives. It hosts the **PEDRO ground stations** (in partnership with PhilSA) and has spearheaded projects to leverage satellite data for disaster risk reduction and other applications. Key ASTI initiatives include:
* **DATOS Help Desk** – *Remote Sensing and Data Science Help Desk*, which is the Philippines’ first AI-assisted system for rapid disaster mapping. The DATOS project was established to produce and disseminate critical maps (e.g. flood extent, landslide maps) to government responders during events[[20]](https://philsa.gov.ph/news/philippinesatellitewatch-wrath-of-typhoon-odette-as-seen-from-space/#:~:text=space%20philsa.gov.ph%20%20The%20DOST,to%20national%20agencies%20with)[[21]](https://archive.opengovasia.com/2024/02/29/tech-solutions-for-disaster-risk-reduction-management-in-the-philippines/#:~:text=The%20Remote%20Sensing%20and%20Data,enhancing%20preparedness%20and%20response%20measures). By integrating satellite imagery (like Sentinel-1 for floods) with automation, DATOS can generate maps within hours of a disaster. This service complements traditional agencies by providing up-to-date hazard impact insights. For example, after Typhoon *Odette* (Rai) in 2021, DATOS provided flood maps derived from Sentinel-1 to aid response[[22]](https://philsa.gov.ph/news/philippinesatellitewatch-wrath-of-typhoon-odette-as-seen-from-space/#:~:text=PhilippineSatelliteWatch%3A%20Wrath%20of%20Typhoon%20Odette,to%20national%20agencies%20with). The platform uses GIS, remote sensing, and AI techniques to ensure **pertinent disaster information** reaches end-users quickly[[21]](https://archive.opengovasia.com/2024/02/29/tech-solutions-for-disaster-risk-reduction-management-in-the-philippines/#:~:text=The%20Remote%20Sensing%20and%20Data,enhancing%20preparedness%20and%20response%20measures).
* **Philippine Sky Artificial Intelligence Program (SkAI-Pinas)** – This is DOST’s flagship R&D program on AI for Earth observation. SkAI-Pinas (a play on *“Sky”* and *“AI”*, pronounced *“sky pinas”*) aims to *“bridge the gap between the availability of massive remote sensing data in the country and the lack of a sustainable framework to process them”*[[4]](https://www.pna.gov.ph/articles/1136226#:~:text=Philippine%20Sky%20Artificial%20Intelligence%20Program,widespread%20processing%20systematically%20and%20effectively). In essence, while satellites (like Sentinel) provide terabytes of images, there is a need for local expertise, tools, and pipelines to turn those into useful information. SkAI-Pinas addresses this by creating an **AI knowledge base**, developing an **Automated Labelling Machine** for large-scale image annotation, and setting up an **AI model repository**[[4]](https://www.pna.gov.ph/articles/1136226#:~:text=Philippine%20Sky%20Artificial%20Intelligence%20Program,widespread%20processing%20systematically%20and%20effectively)[[23]](https://archive.opengovasia.com/2021/04/13/the-philippines-dost-launches-new-ai-programmes/#:~:text=Artificial%20Intelligence%20Programme%20%28SkAI,and%20present%20remote%20sensing%20projects). The program fosters collaborations between ASTI and universities to build human capacity as well. A concrete output is a repository of pre-trained models and labeled datasets to accelerate remote sensing AI workflows in the Philippines[[24]](https://www.pna.gov.ph/articles/1136226#:~:text=This%20comprises%20an%20AI%20knowledge,and%20present%20remote%20sensing%20projects).
* **DIMER (Democratized Intelligent Model Exchange Repository):** DIMER is an upcoming platform under DOST-ASTI intended to be an online repository of pre-trained AI models and geospatial datasets. As hinted in SkAI-Pinas, sharing models (for instance, a trained convolutional neural network for coconut plantation detection) can greatly speed up project startups. DIMER will allow researchers to publish and reuse models, promoting a **“model commons”** for EO applications. By lowering the barrier to entry (users can fine-tune existing models instead of training from scratch), DIMER aligns with the data-centric paradigm – focusing on reusing and improving existing models with local data.
* **AIPI (AI Processing Interface):** AIPI is an initiative to create an accessible interface or platform where users (even non-programmers) can run AI workflows on EO data. This could be envisioned as a web-based portal where one can upload imagery or select an area, choose from a library of AI models (e.g. building footprint extractor), and obtain results without dealing with complex coding. AIPI essentially would operationalize AI for end-users like local government units, integrating data (potentially from the CopPhil Mirror or NAMRIA) with algorithms from DIMER/SkAI-Pinas. Together, DIMER and AIPI aim to **democratize AI for EO** – providing both the tools (models) and the means to apply them easily.
* **PAGASA (Philippine Atmospheric, Geophysical and Astronomical Services Administration):** PAGASA is the national meteorological agency, and while not primarily an EO agency, it relies on satellite data for weather and climate monitoring. PAGASA operates a **Doppler radar and weather satellite receiving network** (for Himawari-8, for example) to gather real-time weather imagery. They produce datasets like daily rainfall estimates, typhoon tracks, and climate outlooks. For EO professionals, PAGASA’s data can complement Sentinel data in applications such as flood models (using rainfall as input) or crop monitoring (correlating weather patterns with Sentinel-derived indices). Moreover, PAGASA’s archives of historical tropical cyclone paths and rainfall can be used in AI models to predict disaster impacts or to help annotate events in satellite images (e.g., identifying images “before” and “after” a typhoon). Integration of **weather, climate, and satellite data** is a rich area for AI – for instance, training a model to predict rice yield might combine Sentinel-2 vegetation indices with PAGASA’s seasonal rainfall anomalies as features.

Each of these agencies contributes pieces of the EO puzzle. **Local datasets** from these sources complement Sentinel data by adding context or ground truth. For example, a **hazard map** from NAMRIA (showing flood-prone zones) used alongside Sentinel-1 flood detections can improve an AI model’s accuracy by providing prior knowledge of likely flood areas. Similarly, **land cover maps** or agricultural parcel data can serve as training labels for supervised classification of Sentinel-2 images. The Philippine EO ecosystem also means there are local platforms to access data: e.g., NAMRIA’s Geoportal for base maps, PhilSA’s Space Data Dashboard for recent satellite analyses, and DOST-ASTI’s upcoming systems like DIMER/AIPI for AI workflows. As an EO professional in the Philippines, being aware of these resources helps you obtain **auxiliary data** (like DEMs, hazard maps, socio-economic data) and **domain expertise** that can significantly enhance machine learning models built on Sentinel imagery.

**Activity – CopPhil Mirror Site & Digital Space Campus:** A recent milestone in the PH-EU space cooperation is the **Copernicus Philippine Mirror Site (CopPhil)** and its accompanying **Digital Space Campus**. CopPhil is a collaborative program with the EU to strengthen Philippine capacity in EO, and it delivers concrete tools for data access and training[[25]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=ESA%20and%20the%20European%20Commission,food%20security%20and%20environmental%20protection)[[26]](https://copphil.philsa.gov.ph/#:~:text=The%20objective%20of%20CopPhil%20,open%20Copernicus%20Earth%20Observation%20data).

* The **CopPhil Copernicus Mirror Site** is a local data hub – essentially a copy of the Copernicus Sentinel archive, hosted in the Philippines (the first of its kind in Asia)[[25]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=ESA%20and%20the%20European%20Commission,food%20security%20and%20environmental%20protection). With €7.3 million support from the EU, this mirror site allows users in the region to download Sentinel-1, -2 (and other Sentinel) data with higher bandwidth and reliability, reducing dependence on European servers[[25]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=ESA%20and%20the%20European%20Commission,food%20security%20and%20environmental%20protection)[[27]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=uptake%20of%20the%20Copernicus%20Programme,Earth%20observation%20programme%20to%20date). The mirror site is part of a broader “National Copernicus Capacity Support Program” aimed at boosting the Philippines’ resilience to natural disasters and climate change through space data[[28]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=Asia,food%20security%20and%20environmental%20protection). By having a local repository, agencies like DOST, PhilSA, PAGASA, and even universities can more quickly access imagery for time-critical applications (e.g., retrieving a Sentinel-1 scene immediately after a typhoon hits). The CopPhil Centre also develops **pilot services** in thematic areas like land cover mapping, disaster monitoring, and marine habitat mapping[[29]](https://copphil.philsa.gov.ph/#:~:text=As%20one%20of%20the%20world%E2%80%99s,data%20in%20three%20thematic%20areas)[[30]](https://copphil.philsa.gov.ph/#:~:text=,benthic%29%20Habitat%20Monitoring%20Service), showcasing how combining Sentinel data with local needs can yield operational products.
* The **CopPhil Digital Space Campus** is an e-learning and capacity-building platform under the CopPhil program[[31]](https://courses.copphil.philsa.gov.ph/#:~:text=,under%20the%20ESA%20CopPhil%20Project)[[32]](https://courses.copphil.philsa.gov.ph/#:~:text=,Benthic%20Habitat%20Monitoring). Delivered in partnership with international and local universities, this online campus provides **training courses on EO and AI** tailored to the Philippine context. Topics include how to use Copernicus data for disaster risk reduction, agriculture, coastal monitoring, etc. The Digital Space Campus contains interactive courses, tutorials, and even recorded webinars, enabling professionals to upskill in remote sensing and machine learning techniques[[33]](https://courses.copphil.philsa.gov.ph/#:~:text=Through%20a%20mix%20of%C2%A0online%20and,policy%20implementation%20in%20the%20Philippines)[[34]](https://courses.copphil.philsa.gov.ph/#:~:text=Skip%20course%20categories). For example, one course might walk through using Sentinel-2 for land cover classification in Palawan, while another might demonstrate ground motion monitoring with Sentinel-1 InSAR – all with local case studies. The platform is essentially an **online training hub** that ensures the knowledge transfer component of CopPhil: it doesn’t suffice to have data (Mirror Site) if users aren’t trained to use it, hence the Digital Campus fills that gap.

Together, these initiatives indicate a strong support system for anyone working in AI/ML for EO in the Philippines. You have local data repositories, organizations eager to collaborate, and growing human capital. As we move to Session 2, keep in mind how these Philippine-specific resources (data and expertise) can be leveraged alongside global tools. For instance, you might source your satellite imagery from the CopPhil Mirror, use a DOST-provided training dataset (labels), run an analysis in Google Earth Engine, and validate results with NAMRIA maps – a workflow that blends global and local strengths.

## Session 2: Core Concepts of AI/ML for Earth Observation

### Module 1: What is AI/ML? The AI/ML Workflow in EO

**Artificial Intelligence (AI)** broadly refers to the capability of machines to perform tasks that typically require human intelligence, such as reasoning or decision-making. In practice, AI often involves computers **simulating human thought processes** via mathematical models[[35]](https://earthi.space/capabilities/ai-and-machine-learning/#:~:text=Artificial%20intelligence%20refers%20to%20the,and%20flexibility%20of%20human%20thinking). **Machine Learning (ML)** is a **subset of AI** – it specifically denotes algorithms that enable computers to *learn from data* and improve performance on a task without being explicitly programmed for every scenario[[36]](https://earthi.space/capabilities/ai-and-machine-learning/#:~:text=Machine%20learning%20is%20a%20subset,computer%20learn%20without%20direct%20instruction). In other words, instead of coding a set of fixed rules, we provide an ML algorithm with example data and it “figures out” the patterns or rules. A classic analogy: rather than telling a program “if pixel is green then forest,” we feed an ML model many satellite images labeled as forest or non-forest and let it **learn the spectral patterns** of forests on its own. This learned model can then predict new, unseen images.

In Earth Observation, AI/ML techniques have become essential to handle the **big data** from satellites and extract actionable information. For context, a single Sentinel-2 scene (100 km²) has millions of pixels across 13 bands – far too much for manual analysis, but amenable to automated pattern recognition by ML. Before diving into specific ML types, let’s outline a **typical AI/ML workflow in EO**, from problem definition to deployment. Figure 2 provides a high-level flowchart of the steps involved in building an ML solution for an EO problem.

[[37]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=machine%20learning%20model)[[38]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=2)A **machine learning workflow** generally includes the following stages:

1. **Problem Definition:** Clearly define the EO problem and desired outcome. In this stage, you ask: *What decision or prediction do I need to make?* For example, *“Classify land cover types from satellite images,”* or *“Predict rice crop yield from multi-temporal data.”* The problem definition sets the target variable (classification categories or regression value) and scope (spatial/temporal extent, resolution, etc.). A well-defined problem also involves understanding user requirements – e.g., an agency might need a forest cover map with 90% accuracy updated annually.
2. **Data Acquisition:** Identify and gather the relevant data required. For EO tasks, this typically means satellite imagery (such as Sentinel-1 SAR or Sentinel-2 optical) and often **auxiliary datasets**. Auxiliary data could be ground truth labels (for supervised learning), digital elevation models, climate data, or existing maps. Data acquisition in practice might involve downloading images from the Copernicus Open Hub or pulling them via an API (as with Google Earth Engine or the CopPhil Mirror). It also includes assembling training data – e.g., collecting sample points of different land cover types from field surveys or higher-resolution images to use as labels.
3. **Data Pre-processing:** Raw satellite data usually need preparation before they can be ingested by an ML model[[39]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=Once%20you%20have%20data%20it,Preprocessing%20involves). Pre-processing steps may include:
4. *Geometric corrections:* aligning images to a map projection (orthorectification) and co-registering multi-date images.
5. *Radiometric corrections:* converting DN values to physical units (reflectance for optical, backscatter coefficient for SAR) and correcting atmospheric effects (for optical, using Level-2A or atmospheric correction models).
6. *Cloud masking:* for optical images, detecting and masking clouds and shadows (using Sentinel-2 QA bands or algorithms like Sen2Cor or QA60 mask).
7. *Cropping/resampling:* clipping imagery to the Area of Interest and ensuring consistent spatial resolution across bands or data sources.
8. *Normalization:* if using multi-date or multi-sensor data, ensuring they are normalized (e.g., normalized difference indices) so that values are comparable.
9. *Feature engineering:* sometimes considered a separate step, but often part of pre-processing in EO – deriving additional layers such as vegetation indices (NDVI, EVI), water indices (NDWI), texture measures, or topographic features (slope, aspect from DEM) that could serve as input features to the model.
10. **Feature Engineering:** Especially for classical ML (and even for deep learning with tabular outputs), selecting and crafting the right features is crucial. In EO, features are often the spectral bands themselves and indices. For instance, a flood detection model might use not just raw SAR backscatter but also the change in backscatter between two dates as a feature, or a textural feature capturing speckle characteristics. Feature engineering may also involve *dimensionality reduction* (like applying PCA to hyperspectral data) or *data augmentation* (creating additional training samples through transformations). The goal is to present the model with informative inputs that correlate well with the target phenomenon.
11. **Model Selection and Training:** Choose an appropriate ML algorithm and train it on the prepared data. The choice of model depends on the task and data size:
12. For a supervised classification of land cover, you might choose algorithms like **Random Forest, Support Vector Machine (SVM), or a Neural Network**. Random Forests are popular for EO classification because they handle high-dimensional data and are relatively robust to noise[[40]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=The%20most%20common%20supervised%20classification,methods%20include). Neural networks (including deep learning models) can be powerful if a large labeled dataset is available.
13. For unsupervised tasks like clustering, you might use **K-means** or other clustering algorithms to group pixels with similar spectral signatures (useful for exploratory analysis or segmentation)[[41]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=Aspect%20Supervised%20Classification%20Unsupervised%20Classification,Minimal%20user%20intervention%20during%20classification).
14. For time series predictions (say, forecasting drought based on NDVI trends), maybe a **recurrent neural network (RNN)** or simpler regression models could be chosen.

During training, the model learns by adjusting its parameters to minimize error on the training data. This entails feeding training examples (e.g., pixel values with known class labels) and using an optimization process (like gradient descent) to improve the model’s predictions. We typically split data into a **training set and validation set** to tune the model and avoid overfitting. For example, out of 1,000 labeled points, 800 might train a model and 200 validate its performance.

1. **Model Evaluation (Validation):** After training, assess the model’s performance on independent **validation or test data**. Common metrics in EO tasks include **overall accuracy, precision/recall, F1-score** for classification, or **RMSE (Root Mean Square Error)** and R² for regression (e.g., biomass estimation). If the model is unsupervised, evaluation might involve comparing clusters to known classes or checking metrics like cluster purity. It’s important to ensure the evaluation data was not seen by the model during training (to truly test generalization). Techniques like cross-validation can also be applied if data is limited.
2. **Deployment and Inference:** Once satisfied with the model, deploy it to generate outputs for the full area or time of interest. In practice, this could mean running the model on every pixel of a Sentinel-2 image to produce a classification map, or setting up a pipeline that automatically processes new Sentinel-1 acquisitions for near-real-time flood detection. Deployment considerations include computational efficiency (using GEE or cloud computing for large areas), and packaging the results for end-users (e.g., a web map service or an analytic report). For operational use, one might integrate the model into an existing workflow – for instance, NAMRIA could incorporate an AI model into its annual land cover mapping process, using it to pre-label satellite images which are then reviewed by analysts.
3. **Monitoring and Maintenance:** (Often overlooked in one-off projects, but critical for operationalization.) Monitor the model’s performance over time – if input data characteristics change (say a new satellite sensor or land cover changes), the model may need retraining or recalibration. Maintenance also involves updating the model when new training data becomes available (continual learning) and ensuring data pipelines (e.g., fetching new imagery) remain functional.

[[42]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=Image%3A%20Flowchart)[[43]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=There%20are%20many%20types%20of,Widely%20used%20are)Figure 2 encapsulates this workflow. Starting from data collection, through cleaning (pre-processing), model training, and evaluation, it emphasizes an iterative approach: if evaluation shows issues, one might loop back to collect more data or engineer better features. In EO projects, this loop is common – for example, if a classifier confuses oil palm plantations with natural forest, you might collect more training samples or add a red-edge band to help differentiation, then retrain.

**Figure 2 – Typical AI/ML Workflow for Earth Observation:** *This flowchart outlines the process of developing an AI/ML solution with EO data. It begins with defining the problem and collecting relevant data (satellite imagery and ground truth). Data is then pre-processed (e.g., atmospheric correction, cloud masking) and features are engineered (spectral indices, etc.). Next, an ML model is selected and trained on the data. The model is evaluated against validation data, and if performance is satisfactory, it is deployed to produce results (maps, predictions) for the target scenario. Feedback from evaluation may loop back to earlier steps to improve the solution.*[[37]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=machine%20learning%20model)[[38]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=2)

*Figure 2: Machine Learning workflow for EO applications, illustrated as a flowchart. Key steps include data collection, data preprocessing (cleaning and feature extraction), model selection/training, and evaluation. The workflow is iterative – insights from evaluation can lead to refining data or model for better performance.*

Understanding this workflow provides a roadmap for your projects: no matter the specific EO application – be it detecting illegal fishing boats with SAR or mapping urban growth with multi-temporal optical imagery – the steps of gathering data, preparing it, choosing/training a model, and validating results will guide your approach.

### Module 2: Types of ML – Supervised vs. Unsupervised Learning (EO Examples)

Machine learning methods are commonly categorized by how they **learn from data**. Two primary categories are **supervised learning** and **unsupervised learning**, each with distinct use-cases in EO.

**Supervised Learning** involves training a model on input data **paired with known output labels**. The algorithm learns to predict the correct output from the inputs. In EO, supervised learning is widely used for tasks like **classification** and **regression**: - In **classification**, the model assigns each input (e.g., each pixel or image) to a discrete category. For example, given a Sentinel-2 pixel’s reflectance values, classify it as **forest, water, urban, or agriculture**. This requires a labeled dataset where many pixels of known land cover type are provided. A practical example is land cover mapping: analysts prepare training polygons for different land cover types from field surveys or high-res imagery, then train a classifier (like Random Forest or a CNN) on Sentinel data to map those classes across the whole region. - In **regression**, the model predicts a continuous numerical value from inputs. For instance, estimating **biomass or crop yield** from satellite indices would be a regression task – you train the model using sample plots with measured biomass as labels and satellite-derived predictors (NDVI, SAR backscatter, etc.) as inputs. Another example: predicting sea surface temperature or soil moisture from satellite microwave readings.

Supervised learning tends to yield high accuracy when **abundant and representative training data** are available[[41]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=Aspect%20Supervised%20Classification%20Unsupervised%20Classification,Minimal%20user%20intervention%20during%20classification)[[44]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=discovered%20from%20data%20patterns%20Accuracy,Suitable%20for%20exploratory%20data%20analysis). The model effectively interpolates patterns it has seen in the labeled data. However, it can struggle to generalize to novel conditions beyond the training set (e.g., a land cover classifier trained only on Luzon may misclassify features in Mindanao if certain land cover types look different there). Thus, careful curation of training data (ensuring all classes and variability are covered) is crucial.

EO Example – *Land Cover Classification (Supervised):* Suppose we want to map different land cover types in Laguna province using Sentinel-2. We gather training polygons for classes: water, built-up, rice paddies, forest, etc., from a combination of field GPS points and existing maps (NAMRIA land cover). Each polygon yields many pixel samples with known class. We compute spectral features for each pixel (raw bands, NDVI, etc.). Using this labeled dataset, we train a **supervised classifier**. A Random Forest classifier might learn, for instance, that **forest pixels** have high NIR and red-edge reflectance, **water pixels** have very low reflectance in NIR (and high in blue for clear water), etc. After training, we apply the model to the whole Sentinel-2 image to produce a thematic map. The accuracy is evaluated against a validation set of labeled pixels. This is a classic supervised workflow resulting in a useful GIS product. Many operational products (like ESA’s global land cover maps or crop type maps) are produced this way, often using advanced models and big training datasets.

EO Example – *Biomass Estimation (Supervised Regression):* We might have field data from sample plots in a forest (from DENR or research plots) that give above-ground biomass in tons per hectare. We want to predict biomass everywhere in that forest using satellite data. We can use Sentinel-1 SAR (sensitive to forest structure) and Sentinel-2 (sensitive to vegetation greenness) as inputs. For each sample plot, derive features: e.g., mean VH and VV backscatter from Sentinel-1, and NDVI and other indices from Sentinel-2. These paired with the field-measured biomass form the training set. We then train a regression model – perhaps an **SVR (Support Vector Regression)** or a **neural network** – that learns the relationship between those satellite features and biomass. Once trained, we apply it to all pixels to get a continuous biomass map. The model is supervised because it used known target values (biomass) to learn. Achieving good results requires careful calibration (biomass signals can saturate at high values for certain sensors) and often a substantial amount of training data covering low to high biomass conditions.

**Unsupervised Learning**, on the other hand, deals with **unlabeled data**. The goal is to find inherent patterns or groupings in the input data without any ground truth to guide the training. The algorithm tries to organize the data in some way – e.g., by similarity. Common unsupervised techniques in EO include **clustering** and **dimensionality reduction**: - **Clustering:** Grouping pixels or objects into clusters based on their feature similarity. In remote sensing, *unsupervised classification* is essentially clustering pixel spectra into a number of clusters, which the analyst then interprets as land cover types[[45]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=The%203%20most%20common%20remote,sensing%20classification%20methods%20are)[[41]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=Aspect%20Supervised%20Classification%20Unsupervised%20Classification,Minimal%20user%20intervention%20during%20classification). K-means is a popular method: you specify *k* (number of clusters), and it will partition the pixels into *k* clusters such that pixels in a cluster are more similar to each other (in spectral feature space) than to those in other clusters. Initially, these clusters have no labels – you must post-assign meaning by comparing with known references or examining the cluster centers. Unsupervised classification is useful when you lack training data and want a first-cut map. It may reveal *natural groupings* in the data (e.g., perhaps it finds clusters corresponding to “bright urban,” “dark vegetation,” “water,” etc.). - **Dimensionality Reduction:** Techniques like PCA (Principal Component Analysis) or t-SNE can be applied to compress high-dimensional data (like multispectral images) into fewer dimensions while preserving most variance or structure. While not a map product per se, this helps in exploratory analysis or as a pre-processing step before using other ML methods.

Unsupervised learning in EO is often used for *exploration* or *initial mapping*. For example, you might run unsupervised clustering on a new satellite image of a remote area to see what distinct spectral classes are present, then go into the field (or use Google Earth) to identify what those classes likely are, then use that information to perform supervised classification later with labels.

EO Example – *Image Clustering (Unsupervised):* We have a Sentinel-2 image of an area with unknown land cover (no maps or labels available). Using an unsupervised method like *K-means clustering*, we ask the algorithm to partition the pixels into, say, 5 clusters based on their 13-band reflectance values. After running it, we get 5 clusters – we examine their spectral signatures (the mean reflectance in each band): - Cluster 1 might have very high NDVI, indicating dense vegetation. - Cluster 2 might have high reflectance in SWIR and low in NIR – possibly urban or bare soil. - Cluster 3 has very low reflectance across all bands – likely water or shadow. - And so on. We then interpret these: Cluster 1 = Forest, Cluster 2 = Urban/Bare, Cluster 3 = Water, etc. This becomes a quick map with classes inferred. It won’t be as accurate as a supervised classification with proper training data, but it can be generated with **no prior labeled data**. It’s also useful to discover subclasses – maybe the clustering separates two kinds of agriculture (because of spectral differences), alerting the analyst to a heterogeneity that a supervised scheme with too few classes might miss[[46]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=What%20are%20the%20main%20differences,as%20we%20classify%20in%20ArcGIS)[[47]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=algorithms%20to%20group%20pixels%20User,Suitable%20for%20exploratory%20data%20analysis).

Another unsupervised approach in EO is **anomaly detection**, which is like clustering where one cluster is “everything normal” and outliers are flagged as anomalies. For instance, an unsupervised model could scan satellite images to find unusual changes (like a new burn scar or flooding) without explicit training – useful for alert systems.

In practice, **semi-supervised learning** and **hybrid approaches** are also employed in EO. Semi-supervised uses a mix of labeled and unlabeled data (common when labels are scarce – you might cluster first, label a few points in each cluster, then refine with supervised learning). And increasingly, **self-supervised learning** (a form of unsupervised pre-training, e.g., training a model to predict missing pixels) is used to leverage large unlabeled datasets before fine-tuning on small labeled sets – very relevant given the huge volume of unlabeled EO imagery.

To summarize: - Supervised learning in EO yields specific, accurate models but needs *ground truth labels*. It shines in well-studied areas with reference data (e.g., crop classification in known agricultural regions). - Unsupervised learning needs no labels and can reveal patterns in new datasets, but the results require interpretation and may be less aligned with real-world categories. It’s great for initial exploration or when automating clustering for quick analyses (like distinguishing water from land automatically in daily images for monitoring). - Many EO workflows actually combine both: e.g., use unsupervised clustering to inform the design of a supervised classification (decide how many classes or to identify mislabeled samples).

### Module 3: Introduction to Deep Learning (Neural Networks, Activation Functions, Loss, Optimizers)

**Deep Learning** is a subset of machine learning that uses multi-layered neural networks to learn complex patterns from data. In EO, deep learning – especially convolutional neural networks (CNNs) – has revolutionized tasks like high-resolution image classification, object detection (e.g., detecting ships or buildings), and segmentation (e.g., creating per-pixel maps of land cover or flooding) due to its ability to automatically learn relevant features from raw data.

At the core of deep learning are **Artificial Neural Networks (ANNs)**. It’s helpful to recall the structure of a basic neural network: - It consists of layers of interconnected *neurons* (also called nodes). - **Input layer**: neurons that take in the input features (e.g., pixel values from different bands). - **Hidden layers**: one or more layers where intermediate computations happen. Each neuron in a hidden layer computes a weighted sum of the outputs from the previous layer and then applies an **activation function**. - **Output layer**: neurons that produce the final predictions (e.g., one neuron per class in classification, or one neuron outputting a continuous value in regression).

In a **fully connected network** (a common architecture for many MLPs – multi-layer perceptrons), each neuron in one layer is connected to every neuron in the next layer, with an associated weight on each connection[[48]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=Architecture%20of%20a%20simple%20neural,Activation%20functions%20are)[[49]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=of%20the%20input%20parameters,input%20and%20the%20output%20data). The network “learns” by adjusting these weights during training.

[[50]](https://en.m.wikipedia.org/wiki/File:Neural_network.svg#:~:text=Size%20of%20this%20PNG%20preview,2%2C560%20%C3%97%201%2C707%20pixels)[[51]](https://en.m.wikipedia.org/wiki/File:Neural_network.svg#:~:text=Description%20Neural%20network,Author%20Dake%2C%20Mysid%20SVG%C2%A0development)Figure 3 shows a simplified neural network architecture: a 3-layer network with an input layer (say 3 input features), one hidden layer with a few neurons, and an output layer. Information flows from input to output (“forward propagation”), and errors are then propagated backward to update weights (“backpropagation”). In EO terms, imagine a network that inputs Red, NIR, SWIR reflectances (3 inputs) and outputs a single value of probability of forest. The hidden layer allows it to combine those inputs in non-linear ways to fit complex decision boundaries that a simple logistic regression (no hidden layer) could not.

**Activation Functions:** After computing the weighted sum (the linear part), each neuron applies an activation function *f(*) to introduce non-linearity[[49]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=of%20the%20input%20parameters,input%20and%20the%20output%20data). Without activation functions (or using only linear functions), a stack of neurons would collapse to a linear model. Activations enable networks to learn non-linear relationships (which are abundant in EO data – reflectance to class mapping is not linear). Common activation functions include: - **ReLU (Rectified Linear Unit):** Outputs 0 if input is negative, otherwise outputs the input itself. ReLU is simple and effective, helping networks train faster by mitigating vanishing gradients. In EO CNNs, ReLU is widely used in hidden layers[[52]](https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/#:~:text=passed%20through%20an%20activation%20function,include%20ReLU%2C%20sigmoid%20and%20tanh). - **Sigmoid:** Outputs a value between 0 and 1 following an S-shaped curve. It was historically used in early neural nets and is still used for output neurons in binary classification (producing a probability). But in hidden layers, sigmoids can cause saturation (gradients becoming very small for large positive/negative inputs). - **Tanh:** Similar to sigmoid but outputs between -1 and 1. Often preferred over sigmoid for hidden layers when zero-centered outputs are beneficial. - **Softmax:** Used in output layer for multi-class classification; it turns raw scores into probabilities that sum to 1 across classes. In an EO classification network, you might use ReLU on hidden layers and a softmax on the output layer to get class probabilities. Activation choice impacts performance – e.g., a forest/non-forest classifier network might use a sigmoid output to indicate probability of forest.

**Loss Functions:** During training, the network’s predictions are evaluated against true labels using a loss (or **cost**) function[[53]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20machine%20learning%20,output%20of%20some%20loss%20function). The loss quantifies the error – the goal is to minimize this. The choice of loss depends on the task: - For classification, **cross-entropy loss** is common. For binary classification (flood vs no-flood), you’d use binary cross-entropy; for multi-class (land cover with N classes), categorical cross-entropy (or equivalently, negative log-likelihood). This loss penalizes wrong confident predictions heavily and is minimized when the model’s predicted probability distribution matches the true distribution (ideally 1 for correct class, 0 for others). - For regression, typical losses are **Mean Squared Error (MSE)** or **Mean Absolute Error (MAE)** – e.g., if predicting temperature or biomass, you measure the average squared difference between predicted and actual values[[54]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20simple%20terms%2C%20a%20loss,inaccurate%2C%20the%20loss%20is%20large). - For segmentation maps, sometimes specialized losses like **Dice coefficient** or **Intersection-over-Union (IoU) loss** are used to directly optimize for overlap of predicted vs true masks (useful in imbalanced cases like small flooded areas vs large non-flood areas). The loss function serves as the **feedback signal** to adjust the network. For example, if our network predicts 0.8 probability for “forest” on a pixel that is actually “non-forest,” the cross-entropy loss might be high; the training process will then nudge the weights to lower that prediction next time[[55]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20simple%20terms%2C%20a%20loss,inaccurate%2C%20the%20loss%20is%20large).

**Optimizers:** To minimize the loss, we use optimization algorithms that iteratively update the network’s weights. The fundamental algorithm is **Gradient Descent** – it computes the gradient of the loss with respect to each weight (via backpropagation) and moves the weights in the opposite direction of the gradient (downhill on the loss surface)[[56]](https://www.ibm.com/think/topics/loss-function#:~:text=Optimization%20algorithms%20such%20as%20gradient,a%20derivative%20at%20all%20points)[[57]](https://www.ibm.com/think/topics/loss-function#:~:text=Using%20the%20gradient%20of%20the,gradient%20and%20thereby%20reduce%20loss). In practice, we use variants: - **Stochastic Gradient Descent (SGD):** Rather than using the entire dataset to compute an exact gradient (which is computationally heavy for big data), SGD uses a random subset (batch) of data each iteration to approximate the gradient, updating weights more frequently. This is almost always used in deep learning training. - **Adaptive methods:** Algorithms like **Adam (Adaptive Moment Estimation)**, RMSProp, Adagrad, etc., automatically adjust the learning rate for each weight based on past gradients. Adam is very popular for deep nets as it generally converges faster and requires less tuning of the learning rate. It’s often a good default optimizer for many EO deep learning tasks unless there’s a reason to prefer plain SGD (like when very fine control over training dynamics is needed). - **Learning Rate:** This hyperparameter determines the size of the weight update step. Too high and training diverges; too low and it’s very slow or gets stuck. Often we use learning rate schedules (decreasing it over time) or techniques like cyclical learning rates. For example, training a CNN on satellite imagery might start with a learning rate of 0.001 and then reduce it to 0.0001 after a few epochs to fine-tune.

Putting it together in an **EO context**: imagine we build a CNN to identify *“deforested areas vs intact forest”* from Sentinel-2 patches. We define the network architecture (say a small U-Net or a simple 5-layer CNN). We choose **binary cross-entropy** as loss and **Adam** optimizer. We feed in many training image patches (with labels), and the network initializes with random weights, outputting random guesses. The loss is high initially. The optimizer computes gradients: if a patch of deforestation was predicted as “forest” with high confidence, the loss gradient will push weights to reduce that confidence next time (e.g., adjust filters to detect the brown soil signal of deforested land). Through many iterations, the network’s filters (in convolutional layers) might start highlighting features like *“loss of canopy (brown areas)”* or *“texture of cleared land”*, improving its ability to detect deforestation automatically. The hidden layers’ neurons are learning increasingly abstract representations of the input (maybe one neuron fires on “green vegetation texture”, another on “soil background”) without us explicitly coding those features – **this automatic feature learning is the power of deep learning**.

It’s worth noting some practical considerations for neural networks: - They require **large training data** to generalize well (to avoid overfitting). Augmentation (random flips, rotations, slight spectral shifts) is often employed in EO to artificially boost training sample diversity. - They are computationally intensive. Training a deep network on 10,000 Sentinel-2 image patches might take hours on a GPU. But once trained, applying (inference) can be fast, especially with optimization and modern hardware. - **Architectures:** Many specialized architectures exist. For EO imagery, **Convolutional Neural Networks (CNNs)** are prevalent for spatial data (as they exploit the spatial structure via convolution filters). **Recurrent Neural Networks (RNNs)** or the newer **transformers** can be used for temporal sequence data (e.g., multi-date monitoring). There are also *hybrid architectures* like ConvLSTM that handle spatio-temporal data. We will explore CNNs and an example (U-Net for segmentation) in later sessions.

**Figure 3 – Simplified Neural Network Architecture:** *This diagram depicts a basic feed-forward neural network with an input layer, one hidden layer, and an output layer. Each circle represents a neuron. Arrows indicate connections (with weights) feeding outputs forward. In the hidden layer, each neuron computes a weighted sum of inputs and applies an activation function f(·). The output layer in this example has two neurons (perhaps indicating a binary classification with two output scores). Such an architecture can learn to map input features (e.g., spectral reflectances) to outputs (e.g., class probabilities) by adjusting the weights (θ) on the connections during training*[*[58]*](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=output%2C%20and%20a%20hidden%20layer,of%20limited%20functions%20with%20certain)[*[48]*](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=Architecture%20of%20a%20simple%20neural,Activation%20functions%20are)*.*[[49]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=of%20the%20input%20parameters,input%20and%20the%20output%20data)[[59]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=layer%29,input%20and%20the%20output%20data)

*Figure 3: A simple neural network architecture. The leftmost layer (green) is the input (each node might be a feature like a spectral band or index). The middle layer (blue) is a hidden layer of neurons that combine inputs through weighted connections and apply an activation function (e.g., ReLU). The rightmost layer (yellow) is the output layer (producing, for instance, two class scores). During training, weights θ are adjusted to minimize the error between predicted and true outputs, enabling the network to learn the mapping from inputs to outputs.*

### Module 4: Data-Centric AI in EO

In recent years, there’s a growing recognition that **improving the data** can often boost AI performance more than tweaking models. This is the essence of the **“Data-Centric AI”** approach[[60]](https://www.vanderschaar-lab.com/dc-check/what-is-data-centric-ai/#:~:text=In%20data,data%20used%20by%20ML%20systems). Traditionally, ML practitioners spent a lot of time on model-centric improvements – trying new architectures, tuning hyperparameters – while treating the dataset as fixed. In data-centric AI, we flip the script: we consider the model (especially with deep learning, many architectures are already quite powerful) as largely given, and focus efforts on ensuring the **data is of high quality, adequately labeled, and representative**[[60]](https://www.vanderschaar-lab.com/dc-check/what-is-data-centric-ai/#:~:text=In%20data,data%20used%20by%20ML%20systems).

Andrew Ng, a leading AI figure, articulated this well: *“The dominant paradigm was to download the dataset and focus on improving the code (model). It’s now more productive to hold the model fixed and instead find ways to improve the data.”*[[61]](https://kili-technology.com/data-labeling/earth-observation-data-labeling-guide#:~:text=As%20Andrew%20Ng%20noted%2C%20%E2%80%9CThe,specialized%20domain%20of%20geospatial%20imagery) For EO professionals, this resonates because we often have complex data challenges: clouds, noise, class imbalance, etc., that can’t be fixed by model choice alone. A mediocre model with excellent data may outperform an excellent model with poor data.

Key aspects of data-centric AI in EO include:

* **Data Quality:** Ensure your input data and labels are accurate and consistent. Satellite data can have errors – e.g., geolocation errors (misalignments between layers), radiometric calibration issues, or cloud contamination in “clear” pixels. Label quality is even more crucial: if your training labels (say crop type field data) are noisy or outdated, the model learns wrong mappings. One should invest time in *cleaning labels* – removing or correcting mis-labeled examples. In a land cover dataset, if some “urban” training points actually fall on bare soil, it’s worth correcting those or moving them to a separate class. Consistency in labeling criteria is vital: all labelers should follow the same definitions (what counts as “mangrove” vs “forest”?), otherwise the model gets confused by seeming label randomness. **Data documentation** and verification steps help catch these issues.
* **Sufficient Quantity (and Coverage):** Deep learning models especially crave lots of examples. But beyond sheer quantity, it’s about covering the variability of the task. In EO, that means:
* Spatial coverage: training data across different regions so the model learns various geographies. A building detector trained only on Manila might fail in rural towns unless it has seen examples of those different building materials and layouts.
* Temporal coverage: if the model will be applied year-round, the training data should include different seasons (vegetation changes dramatically between dry and wet season in some areas, affecting pixel values). If we only train on images from July, the model might misclassify January imagery.
* Sensor coverage: If combining data sources or deploying over time, consider sensor differences. For example, Sentinel-2 and Landsat have different band characteristics; a model trained on one might need adaptation for another or multi-sensor training data.
* **Balanced and Diverse Training Set:** EO datasets often suffer from class imbalance. There’s much more “non-event” than “event” pixels (e.g., 99% of pixels are non-flooded, 1% flooded). Similarly, common land covers (forest, water) might dominate rare ones (mangrove, built-up). A data-centric approach would address this: maybe *oversample* minority classes, or use data augmentation to boost them, or intentionally collect more minority class samples (e.g., use stratified sampling to pick training points). Diversity also means capturing different appearances of the same class – e.g., “forest” can be dense dark green or sparse light green; including both prevents the model from biasing to one type.
* **Data Annotation Strategy:** In EO, labeling data can be expensive (field work or manual photo interpretation). A data-centric mindset pushes for smarter labeling:
* Use **active learning**: iteratively let the model identify uncertain areas and have an expert label those. This targets labeling effort to where it most improves the model.
* Use weak labels or proxies initially, then refine. For instance, you might label training data from an existing coarse map (like a 100 m land cover map) to train a first model at 10 m. That model’s outputs can then be corrected by analysts to create a better training set for a second round.
* Crowdsource or involve local experts for labeling but ensure a quality control process (maybe cross-validation among labelers or having a senior analyst review a subset).
* Embrace **open data**: If someone has published a labeled dataset (like the EuroSAT land cover dataset for Sentinel-2 or Radiant Earth’s agriculture dataset), incorporate it or transfer learn from it, adjusting for local conditions.
* **Common EO Data Issues and Remedies:** Let’s list a few typical problems and data-centric solutions:
* *Clouds and Shadows:* For optical data, cloud cover can severely affect models (e.g., a classifier might confuse clouds as built-up due to brightness). Solution: aggressive cloud masking in pre-processing, and perhaps including a cloud probability band as input so the model can learn to ignore cloudy pixels. Also, consider gap-filling cloudy images with data from other dates or sensors (data fusion).
* *Noisy labels from remote sources:* e.g., labeling cropland via MODIS-based map has coarse errors. Solution: use high-resolution reference in a subset to refine labels; or train a model on noisy labels but then manually clean its highest-error predictions for a second training round (sometimes called “distillation” or “self-training”).
* *Changing landscape (label drift):* EO data is temporal – a 2018 land cover map used for labels won’t be 100% accurate for 2023 imagery. Data-centric view would be to update labels (if possible) or at least not trust older labels for areas known to change (urban expansion zones etc. could be omitted or re-labeled using newer imagery).
* *Geographic bias:* Models can latch onto context that doesn’t generalize (e.g., if all “urban” training patches also contain roads that appear as straight lines, the model might think “straight line = urban”). To combat this, provide diverse examples (urban without obvious roads, roads in rural areas as well) so the model learns true defining features (like spectral or textural differences).
* *Resolution mismatch:* Using labels from one resolution for another (e.g., using 30 m land cover to label 10 m pixels) introduces ambiguity – mixed pixels. A data-centric approach might be to only label pure areas or upscale the imagery to match labels. Alternatively, embrace *soft labels* – e.g., a pixel is 70% crop, 30% forest, so training with fractional labels if model can output probabilities (this is advanced, but sometimes done for coarse training data).

Overall, **data-centric AI encourages an iterative loop on data improvements**. If your flood model is missing some flooded areas, instead of immediately tweaking architecture, examine the data: perhaps the training set lacked examples of flooded *vegetated* land (which has a different SAR signal). The fix: add such examples to training (maybe simulate some by flooding a SAR backscatter threshold, or find past events). As another example, if a land cover model often confuses pasture vs shrubland, perhaps the classes are not well-defined or overlap – maybe you need a better class taxonomy or more consistent labeling rules, or to merge them into one class if even human labelers struggle to distinguish.

Data-centric philosophy doesn’t mean models are unimportant – but it recognizes that for many problems, **good data is the foundation**. In EO projects, we frequently operate in data-limited regimes (labels are scarce) or data-noisy regimes. Investing time on data will pay dividends: - It makes your model training more efficient (less garbage in, so less overfitting to noise). - It makes the model more robust (if it has seen varied conditions, it’s less likely to break when encountering a new scene). - It can simplify your model – a simpler model with clean data can outperform a complex model with messy data. For instance, you might not need a 50-layer CNN if a 10-layer CNN given well-preprocessed, augmented data achieves the goal.

A tangible EO example of data-centric improvement: *Cloud detection.* Instead of developing a fancy model from scratch, one might curate a high-quality dataset of cloud vs non-cloud pixels across many Sentinel-2 scenes (ensuring various cloud types, snow differentiation, etc.). With that, even a relatively standard U-Net model can be trained to excellent accuracy. Conversely, without good training data, even the fanciest model will struggle and produce artifacts (e.g., mistaking bright sand for cloud or vice versa).

To conclude, as you embark on AI/ML projects in EO, allocate a significant chunk of your effort to **examining, cleaning, and enriching your data**. Leverage the Philippine EO ecosystem – e.g., use NAMRIA’s accurate maps to label data, use PAGASA’s climate data to stratify samples (maybe ensuring you label both wet and dry season imagery). Use tools like the CopPhil Digital Campus to learn best practices in data preprocessing. By being **data-centric**, you set your model up for success, because a model can only be as good as the data it learns from. As the saying goes in data science: *“Garbage in, garbage out”*. The converse is our goal: *“Quality in, intelligence out.”*[[61]](https://kili-technology.com/data-labeling/earth-observation-data-labeling-guide#:~:text=As%20Andrew%20Ng%20noted%2C%20%E2%80%9CThe,specialized%20domain%20of%20geospatial%20imagery)[[60]](https://www.vanderschaar-lab.com/dc-check/what-is-data-centric-ai/#:~:text=In%20data,data%20used%20by%20ML%20systems)

[[1]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=) [[25]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=ESA%20and%20the%20European%20Commission,food%20security%20and%20environmental%20protection) [[27]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=uptake%20of%20the%20Copernicus%20Programme,Earth%20observation%20programme%20to%20date) [[28]](https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines#:~:text=Asia,food%20security%20and%20environmental%20protection) ESA - ESA and the European Commission uniting on Earth observation for the Philippines

<https://www.esa.int/Applications/Observing_the_Earth/Copernicus/ESA_and_the_European_Commission_uniting_on_Earth_observation_for_the_Philippines>

[[2]](https://dataspace.copernicus.eu/data-collections/sentinel-data/sentinel-1#:~:text=The%20Sentinel,images%20in%20all%20weather%20conditions) Sentinel-1 | Copernicus Data Space Ecosystem

<https://dataspace.copernicus.eu/data-collections/sentinel-data/sentinel-1>

[[3]](https://www.eoportal.org/satellite-missions/copernicus-sentinel-1#:~:text=SM%20mode%20has%20an%2080,m%20x%20100%20m) Copernicus: Sentinel-1 - eoPortal

<https://www.eoportal.org/satellite-missions/copernicus-sentinel-1>

[[4]](https://www.pna.gov.ph/articles/1136226#:~:text=Philippine%20Sky%20Artificial%20Intelligence%20Program,widespread%20processing%20systematically%20and%20effectively) [[24]](https://www.pna.gov.ph/articles/1136226#:~:text=This%20comprises%20an%20AI%20knowledge,and%20present%20remote%20sensing%20projects) DOST introduces 9 AI R&D projects | Philippine News Agency

<https://www.pna.gov.ph/articles/1136226>

[[5]](https://www.sciencedirect.com/science/article/pii/S0034425722003169#:~:text=Effect%20of%20the%20temporal%20baseline,Potin%20et) Sentinel-1 interferometric coherence as a vegetation index for ...

<https://www.sciencedirect.com/science/article/pii/S0034425722003169>

[[6]](https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi#:~:text=Sentinel,bands%20at%2060%20meter%20resolution) [[11]](https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi#:~:text=13%20bands%3A%20442.3%20nm%20,4%20nm%20%28see%20table) Sentinel-2 Multispectral Imager | NASA Earthdata

<https://www.earthdata.nasa.gov/data/instruments/sentinel-2-msi>

[[7]](https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/#:~:text=Dedicated%20to%20supplying%20data%20for,B09%2C%20B10%2C%20B11%20and%20B12) [[8]](https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/#:~:text=versatile%20set%20of%2013%20spectral,B09%2C%20B10%2C%20B11%20and%20B12) Sentinel-2 | Sentinel Hub custom scripts

<https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel/sentinel-2/>

[[9]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=,meter%20pixel%20size) [[10]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=,meter%20pixel%20size) [[14]](https://gisgeography.com/sentinel-2-bands-combinations/#:~:text=B8a%2020%20m%20865%20nm,SWIR) Sentinel 2 Bands and Combinations - GIS Geography

<https://gisgeography.com/sentinel-2-bands-combinations/>

[[12]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=Figure%202%20Sentinel,Atmospheric) [[13]](https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277#:~:text=correction%20was%20conducted%20using%20Sen2Cor%2C,values%2C%20band%20reflectance%20in%20the) ph01.tci-thaijo.org

<https://ph01.tci-thaijo.org/index.php/aer/article/download/257524/173693/988277>

[[15]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=The%20following%20two%20main%20products,26) [[16]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=%2A%20Level,processor%20from%20ESA%27s%20SNAP%20Toolbox) [[17]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=product%20can%20be%20obtained%20from,by%20the%20user%20with%20the) [[18]](https://en.wikipedia.org/wiki/Sentinel-2#:~:text=instructions%20dataspace.copernicus.eu%20.%20%2A%20Level,processor%20from%20ESA%27s%20SNAP%20Toolbox) Sentinel-2 - Wikipedia

<https://en.wikipedia.org/wiki/Sentinel-2>

[[19]](https://www.facebook.com/PhilSpaceAgency/posts/-%F0%9D%97%9C%F0%9D%97%96%F0%9D%97%AC%F0%9D%97%A0%F0%9D%97%9C-%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%81%F0%9D%97%AE-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%80%F0%9D%97%B5%F0%9D%97%AF%F0%9D%97%BC%F0%9D%97%AE%F0%9D%97%BF%F0%9D%97%B1-%F0%9D%97%B9%F0%9D%97%AE%F0%9D%98%82%F0%9D%97%BB%F0%9D%97%B0%F0%9D%97%B5-%F0%9D%97%B1%F0%9D%98%82%F0%9D%97%BF%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B4-%F0%9D%98%81%F0%9D%97%B5%F0%9D%97%B2-%F0%9D%97%A3%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%B9%F0%9D%97%B6%F0%9D%97%BD%F0%9D%97%BD%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B2%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2%F0%9D%97%AA%F0%9D%97%B2%F0%9D%97%B2%F0%9D%97%B8-%F0%9D%97%98%F0%9D%98%85%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%AF%F0%9D%97%B6%F0%9D%98%81the-phi/925350012971936/#:~:text=Philippine%20Space%20Agency%20,2024%20at%20the%20Quantum%20Skyview) Philippine Space Agency - Facebook

<https://www.facebook.com/PhilSpaceAgency/posts/-%F0%9D%97%9C%F0%9D%97%96%F0%9D%97%AC%F0%9D%97%A0%F0%9D%97%9C-%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%81%F0%9D%97%AE-%F0%9D%97%97%F0%9D%97%AE%F0%9D%98%80%F0%9D%97%B5%F0%9D%97%AF%F0%9D%97%BC%F0%9D%97%AE%F0%9D%97%BF%F0%9D%97%B1-%F0%9D%97%B9%F0%9D%97%AE%F0%9D%98%82%F0%9D%97%BB%F0%9D%97%B0%F0%9D%97%B5-%F0%9D%97%B1%F0%9D%98%82%F0%9D%97%BF%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B4-%F0%9D%98%81%F0%9D%97%B5%F0%9D%97%B2-%F0%9D%97%A3%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%B9%F0%9D%97%B6%F0%9D%97%BD%F0%9D%97%BD%F0%9D%97%B6%F0%9D%97%BB%F0%9D%97%B2%F0%9D%97%A6%F0%9D%97%BD%F0%9D%97%AE%F0%9D%97%B0%F0%9D%97%B2%F0%9D%97%AA%F0%9D%97%B2%F0%9D%97%B2%F0%9D%97%B8-%F0%9D%97%98%F0%9D%98%85%F0%9D%97%B5%F0%9D%97%B6%F0%9D%97%AF%F0%9D%97%B6%F0%9D%98%81the-phi/925350012971936/>

[[20]](https://philsa.gov.ph/news/philippinesatellitewatch-wrath-of-typhoon-odette-as-seen-from-space/#:~:text=space%20philsa.gov.ph%20%20The%20DOST,to%20national%20agencies%20with) [[22]](https://philsa.gov.ph/news/philippinesatellitewatch-wrath-of-typhoon-odette-as-seen-from-space/#:~:text=PhilippineSatelliteWatch%3A%20Wrath%20of%20Typhoon%20Odette,to%20national%20agencies%20with) PhilippineSatelliteWatch: Wrath of Typhoon Odette as seen from space

<https://philsa.gov.ph/news/philippinesatellitewatch-wrath-of-typhoon-odette-as-seen-from-space/>

[[21]](https://archive.opengovasia.com/2024/02/29/tech-solutions-for-disaster-risk-reduction-management-in-the-philippines/#:~:text=The%20Remote%20Sensing%20and%20Data,enhancing%20preparedness%20and%20response%20measures) Tech Solutions for Disaster Risk Reduction Management in the Philippines – OpenGov Asia

<https://archive.opengovasia.com/2024/02/29/tech-solutions-for-disaster-risk-reduction-management-in-the-philippines/>

[[23]](https://archive.opengovasia.com/2021/04/13/the-philippines-dost-launches-new-ai-programmes/#:~:text=Artificial%20Intelligence%20Programme%20%28SkAI,and%20present%20remote%20sensing%20projects) The Philippines’ DOST launches new AI programmes – OpenGov Asia

<https://archive.opengovasia.com/2021/04/13/the-philippines-dost-launches-new-ai-programmes/>

[[26]](https://copphil.philsa.gov.ph/#:~:text=The%20objective%20of%20CopPhil%20,open%20Copernicus%20Earth%20Observation%20data) [[29]](https://copphil.philsa.gov.ph/#:~:text=As%20one%20of%20the%20world%E2%80%99s,data%20in%20three%20thematic%20areas) [[30]](https://copphil.philsa.gov.ph/#:~:text=,benthic%29%20Habitat%20Monitoring%20Service) Home - CopPhil

<https://copphil.philsa.gov.ph/>

[[31]](https://courses.copphil.philsa.gov.ph/#:~:text=,under%20the%20ESA%20CopPhil%20Project) [[32]](https://courses.copphil.philsa.gov.ph/#:~:text=,Benthic%20Habitat%20Monitoring) [[33]](https://courses.copphil.philsa.gov.ph/#:~:text=Through%20a%20mix%20of%C2%A0online%20and,policy%20implementation%20in%20the%20Philippines) [[34]](https://courses.copphil.philsa.gov.ph/#:~:text=Skip%20course%20categories) Home | copphildigitalcampus

<https://courses.copphil.philsa.gov.ph/>

[[35]](https://earthi.space/capabilities/ai-and-machine-learning/#:~:text=Artificial%20intelligence%20refers%20to%20the,and%20flexibility%20of%20human%20thinking) [[36]](https://earthi.space/capabilities/ai-and-machine-learning/#:~:text=Machine%20learning%20is%20a%20subset,computer%20learn%20without%20direct%20instruction) AI & Machine Learning | Earth-i

<https://earthi.space/capabilities/ai-and-machine-learning/>

[[37]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=machine%20learning%20model) [[38]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=2) [[39]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=Once%20you%20have%20data%20it,Preprocessing%20involves) [[42]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=Image%3A%20Flowchart) [[43]](https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/#:~:text=There%20are%20many%20types%20of,Widely%20used%20are) Flowchart for basic Machine Learning models - GeeksforGeeks

<https://www.geeksforgeeks.org/machine-learning/flowchart-for-basic-machine-learning-models/>

[[40]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=The%20most%20common%20supervised%20classification,methods%20include) [[41]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=Aspect%20Supervised%20Classification%20Unsupervised%20Classification,Minimal%20user%20intervention%20during%20classification) [[44]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=discovered%20from%20data%20patterns%20Accuracy,Suitable%20for%20exploratory%20data%20analysis) [[45]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=The%203%20most%20common%20remote,sensing%20classification%20methods%20are) [[46]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=What%20are%20the%20main%20differences,as%20we%20classify%20in%20ArcGIS) [[47]](https://gisgeography.com/supervised-unsupervised-classification-arcgis/#:~:text=algorithms%20to%20group%20pixels%20User,Suitable%20for%20exploratory%20data%20analysis) Supervised and Unsupervised Classification in Remote Sensing - GIS Geography

<https://gisgeography.com/supervised-unsupervised-classification-arcgis/>

[[48]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=Architecture%20of%20a%20simple%20neural,Activation%20functions%20are) [[49]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=of%20the%20input%20parameters,input%20and%20the%20output%20data) [[58]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=output%2C%20and%20a%20hidden%20layer,of%20limited%20functions%20with%20certain) [[59]](https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616#:~:text=layer%29,input%20and%20the%20output%20data) Architecture of a simple neural network described by the input, output,... | Download Scientific Diagram

<https://www.researchgate.net/figure/Architecture-of-a-simple-neural-network-described-by-the-input-output-and-a-hidden_fig1_340568616>

[[50]](https://en.m.wikipedia.org/wiki/File:Neural_network.svg#:~:text=Size%20of%20this%20PNG%20preview,2%2C560%20%C3%97%201%2C707%20pixels) [[51]](https://en.m.wikipedia.org/wiki/File:Neural_network.svg#:~:text=Description%20Neural%20network,Author%20Dake%2C%20Mysid%20SVG%C2%A0development) File:Neural network.svg - Wikipedia

<https://en.m.wikipedia.org/wiki/File:Neural_network.svg>

[[52]](https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/#:~:text=passed%20through%20an%20activation%20function,include%20ReLU%2C%20sigmoid%20and%20tanh) What is a Neural Network? - GeeksforGeeks

<https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/>

[[53]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20machine%20learning%20,output%20of%20some%20loss%20function) [[54]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20simple%20terms%2C%20a%20loss,inaccurate%2C%20the%20loss%20is%20large) [[55]](https://www.ibm.com/think/topics/loss-function#:~:text=In%20simple%20terms%2C%20a%20loss,inaccurate%2C%20the%20loss%20is%20large) [[56]](https://www.ibm.com/think/topics/loss-function#:~:text=Optimization%20algorithms%20such%20as%20gradient,a%20derivative%20at%20all%20points) [[57]](https://www.ibm.com/think/topics/loss-function#:~:text=Using%20the%20gradient%20of%20the,gradient%20and%20thereby%20reduce%20loss) What is Loss Function? | IBM

<https://www.ibm.com/think/topics/loss-function>

[[60]](https://www.vanderschaar-lab.com/dc-check/what-is-data-centric-ai/#:~:text=In%20data,data%20used%20by%20ML%20systems) What is Data-Centric AI? // van der Schaar Lab

<https://www.vanderschaar-lab.com/dc-check/what-is-data-centric-ai/>

[[61]](https://kili-technology.com/data-labeling/earth-observation-data-labeling-guide#:~:text=As%20Andrew%20Ng%20noted%2C%20%E2%80%9CThe,specialized%20domain%20of%20geospatial%20imagery) Overcoming Earth Observation Data Labeling Challenges: A Guide for ML Engineers

<https://kili-technology.com/data-labeling/earth-observation-data-labeling-guide>