DEVELOPING THE INTERNET OF NATURE

António Câmara

May 6th, 2025

Internet of Nature (IoN) uses sensors, satellite imagery, and algorithms to transform urban green spaces into data, enhancing our ecosystem management

Nadina Galle

DEVELOPING THE INTERNET OF NATURE
A broader vision

A broader sense of Nature

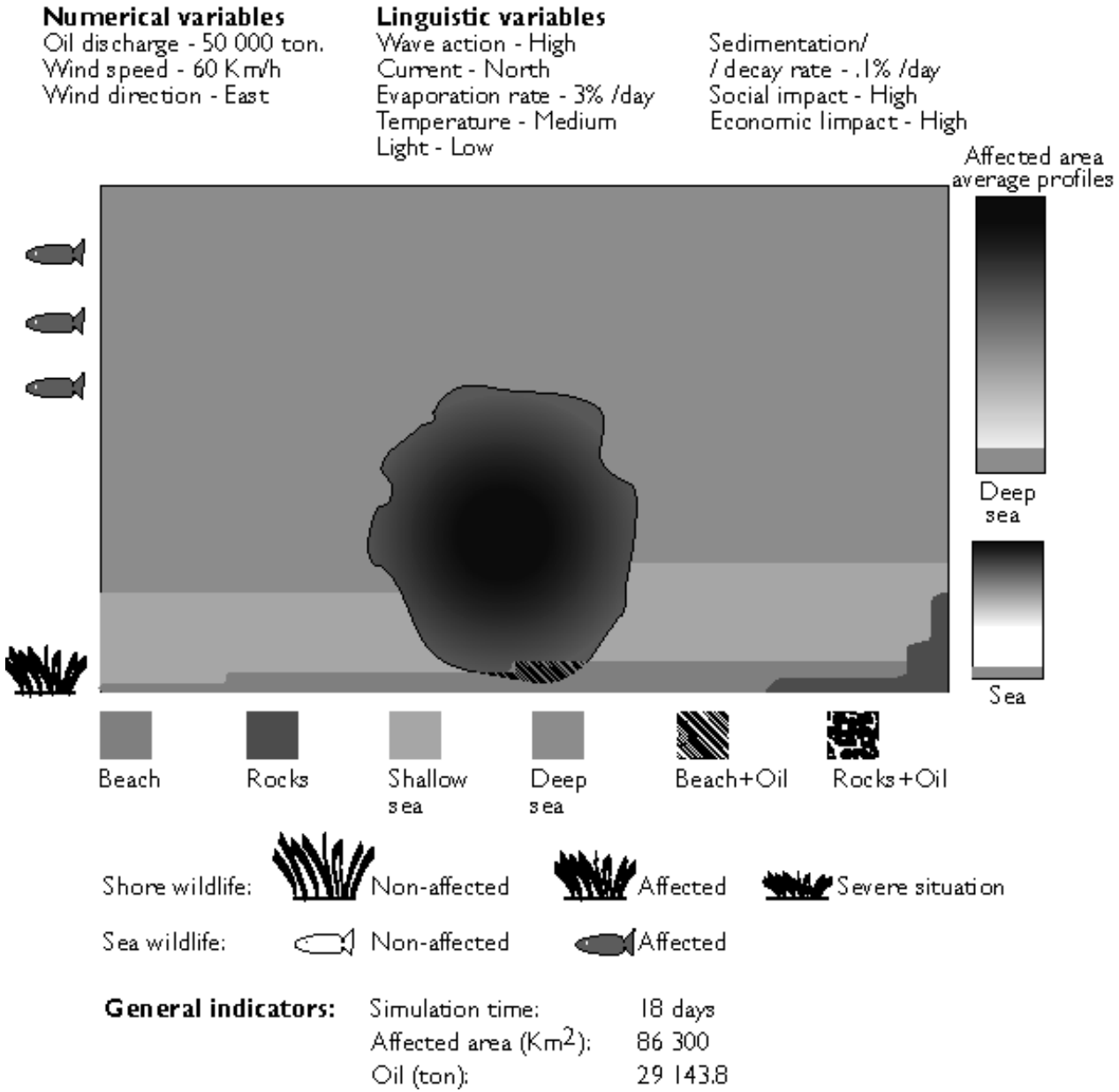
Inspirations: multimodal simulation, spatial data infrastructures, spatial computing, AI software and hardware

Humans, machines and Nature

Internet of Nature: A proposal for disruption

Appendices: Multimodal models, Agents for the Internet of Nature

INSPIRATIONS: Multimodal simulation



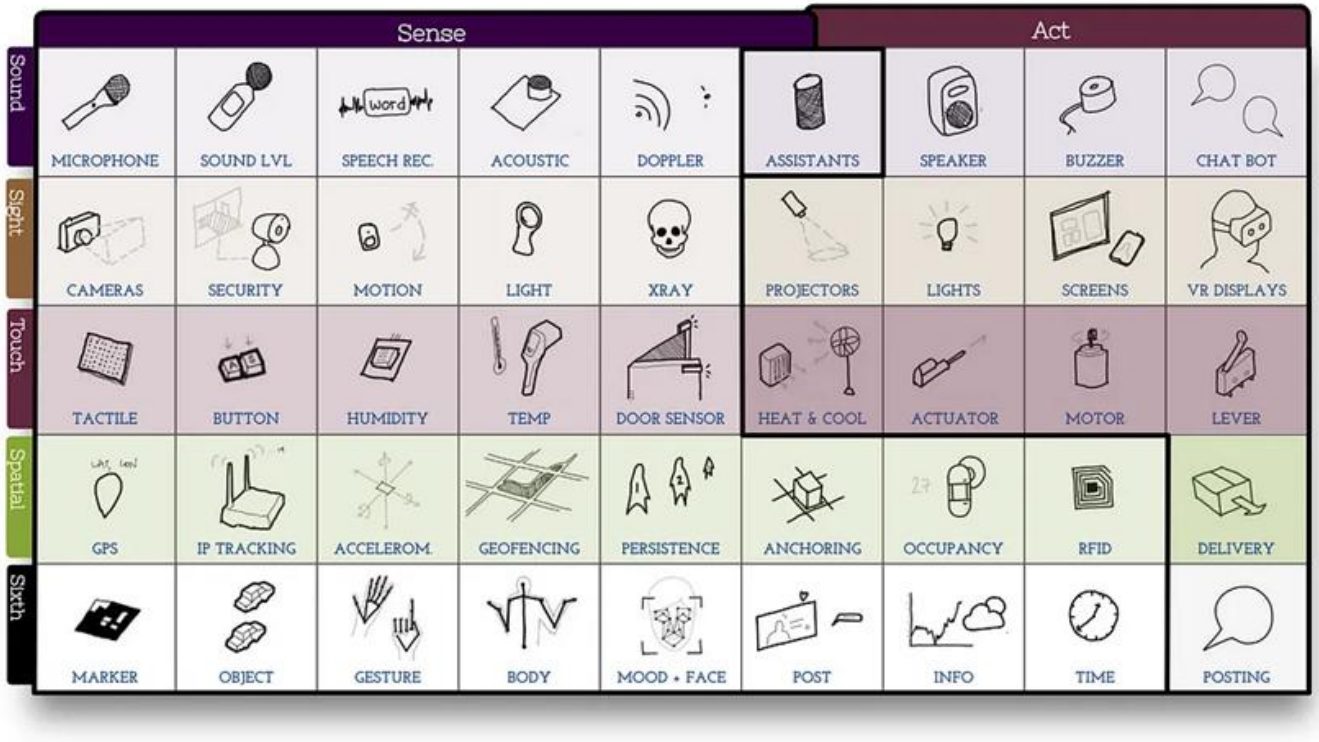
INSPIRATIONS: Spatial data infrastructures

SNIG- The first Web (1995) and Virtual Reality based spatial data infrastructure (1998)

Virtual GIS

INSPIRATIONS: Spatial Computing

YDreams, 2000-2025



Periodic Table of Spatial Computing

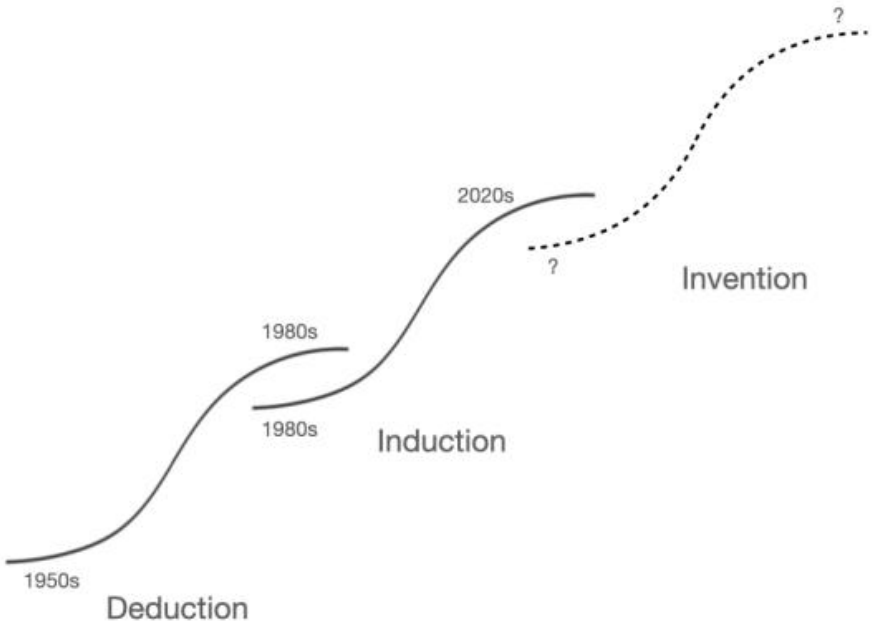
INSPIRATIONS: AI Software And Hardware

Tuga Innovations, 2022-2025

TUGA LAND AIR Combo

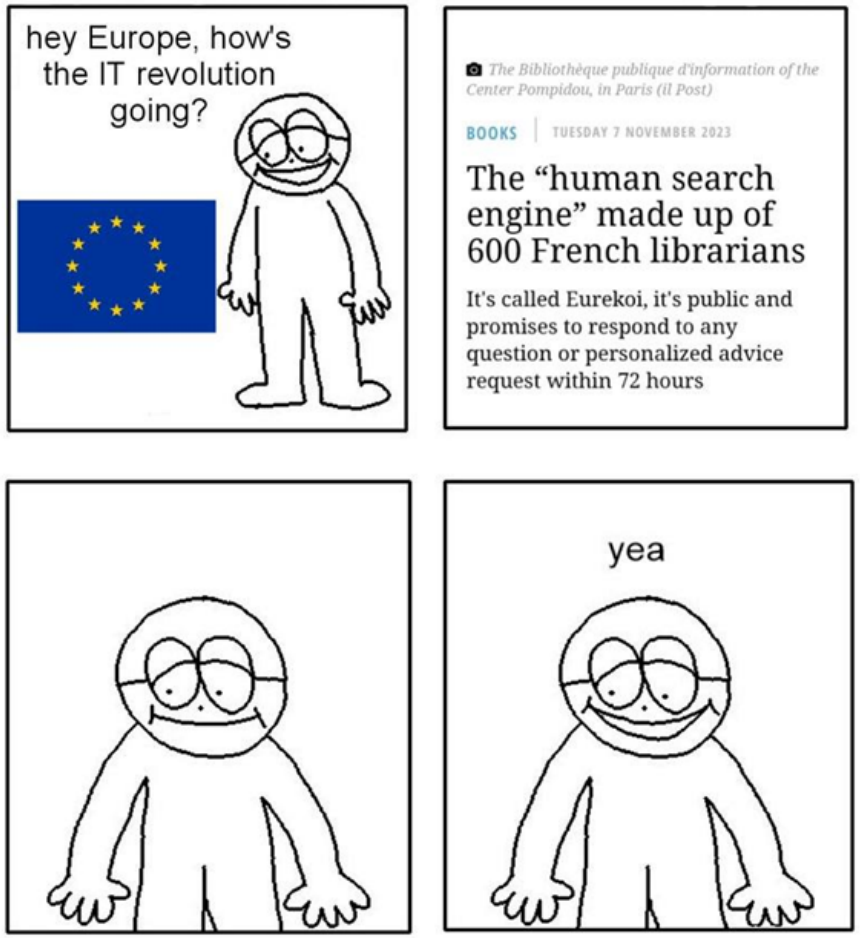
HUMANS, MACHINES AND NATURE

- Humans
- Humans and Machines
- Humans and Nature

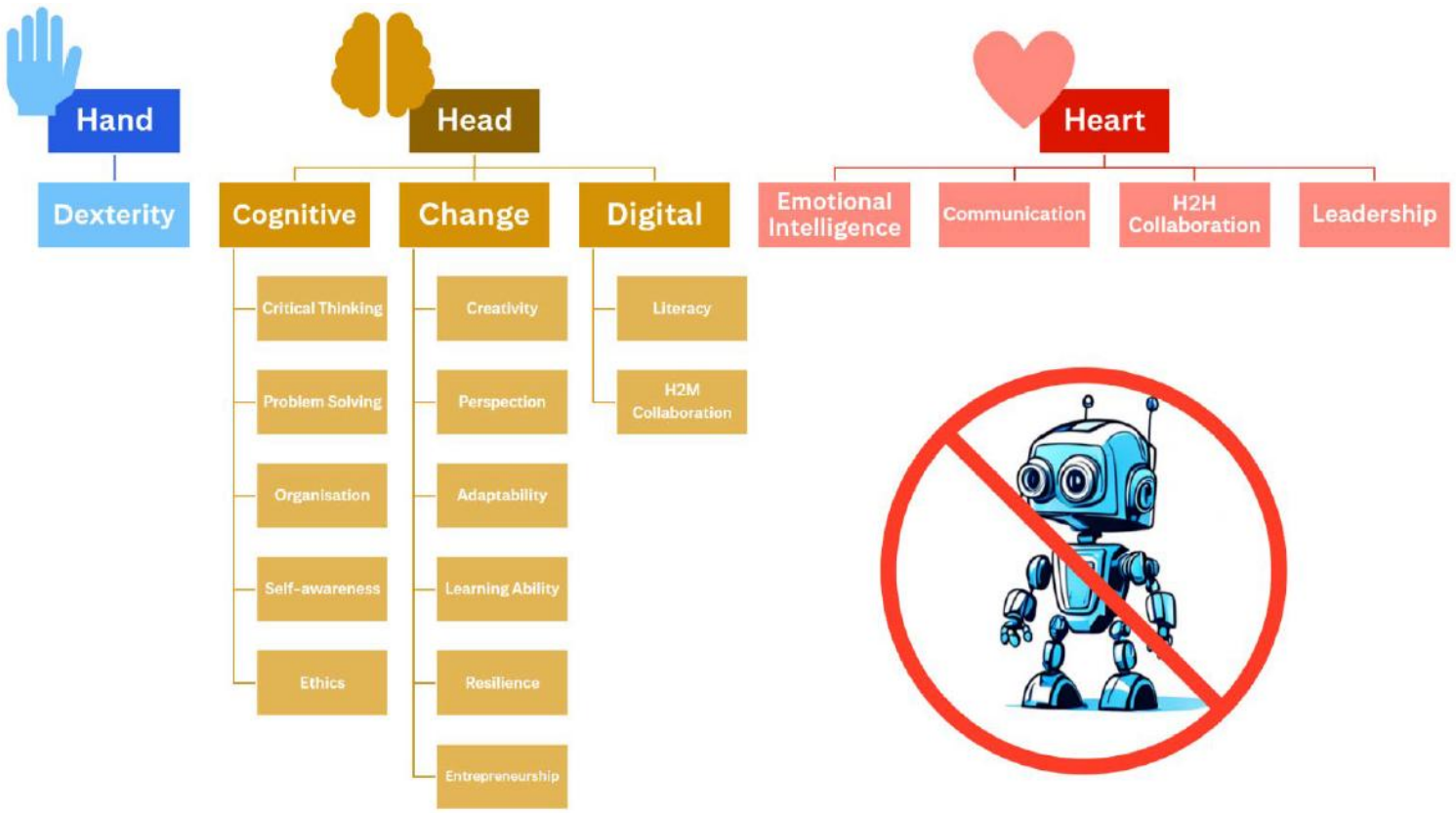


10 Grand Challenges We Will Face in 2050

HUMANS



We must love the French



[What Machines can not master](#)

[The Human Soul](#)

HUMANS AND MACHINES

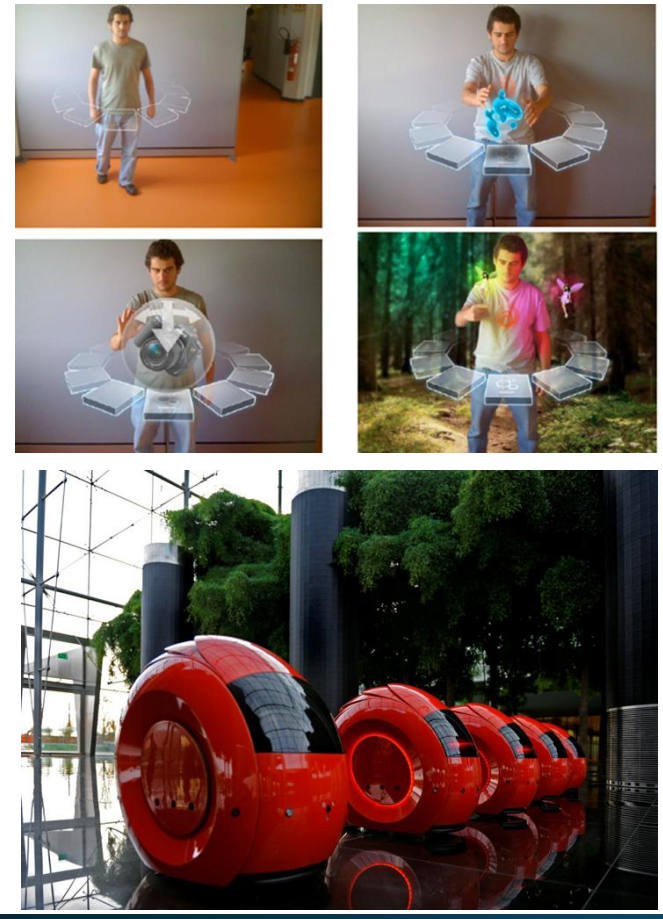
Spatial Computing (AI/Robotics/AR/VR) fundamentals

YVision mathematics

- 3D Rendering: Linear Algebra - Differential Geometry - Geometry (Projective Transformations)
- Physics Simulation: Linear Algebra - Differential Calculus - Integral Calculus - Numerical Analysis (for approximating continuous mathematics)
- Computer Vision: Linear Algebra - Geometry (Homographies, Projective Transformations) - Cellular Automata - Convolutions - Fourier Analysis
- Synthesis and Transform Audio: Fourier Analysis, Synthesis and Transform
- Machine Learning: Statistical Analysis - Artificial Neural Networks - Principal Component Analysis - Regression - Function Approximation
- Evolutionary Computation Core Framework: Lambda Calculus (root of functional programming) - Turing Machines - Theory of Computation

[The YDreams Collection]

Spatial Computing (AI/Robotics/AR/VR) applications



HUMANS AND MACHINES

Spatial tags as atomic entities of the Internet of Nature

HUMANS AND MACHINES

Spatial tags as atomic entities of the Internet of Nature

GEOGRAPHICALLY DISTRIBUTED THROUGHOUT A TERRITORY

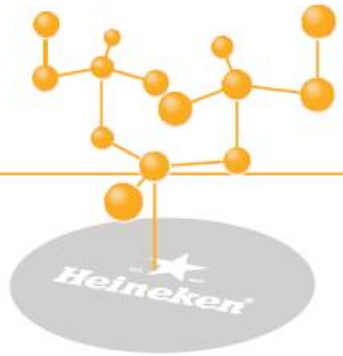
Smart tags are fixed to GPS coordinates. Flexible usage including identification of restaurants and stores, or providing tourist information about places and monuments.

RELATIVE TO A STORE SHELF

Smart tags are assigned positions relative to a store shelf. Since the tags also know their location relative to one another, they can be used to identify where a user is in the store and orient them to other products they are looking for.

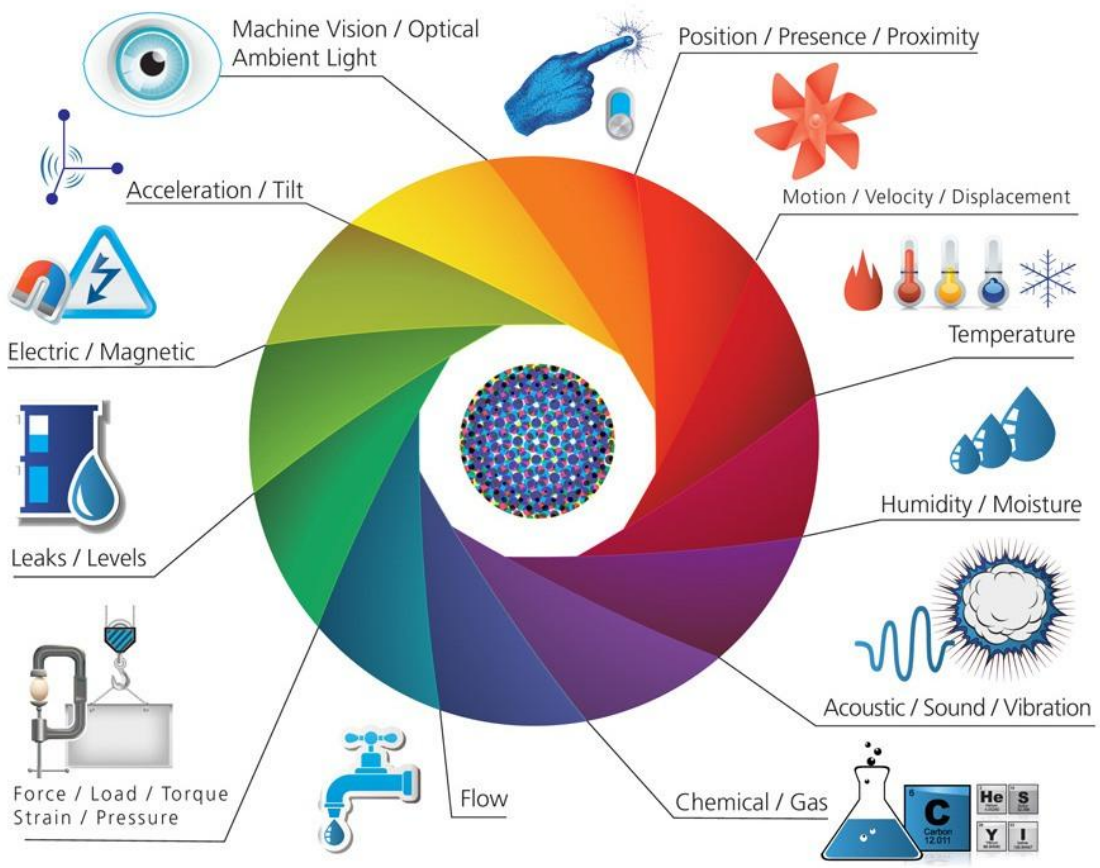
BRAND EXPLORATION

Smart tags can be assigned to a particular brand and organized around it to allow the user to explore the brand's world. A variety of content from the brand may be included and the tags can act as a hierarchical menu system.

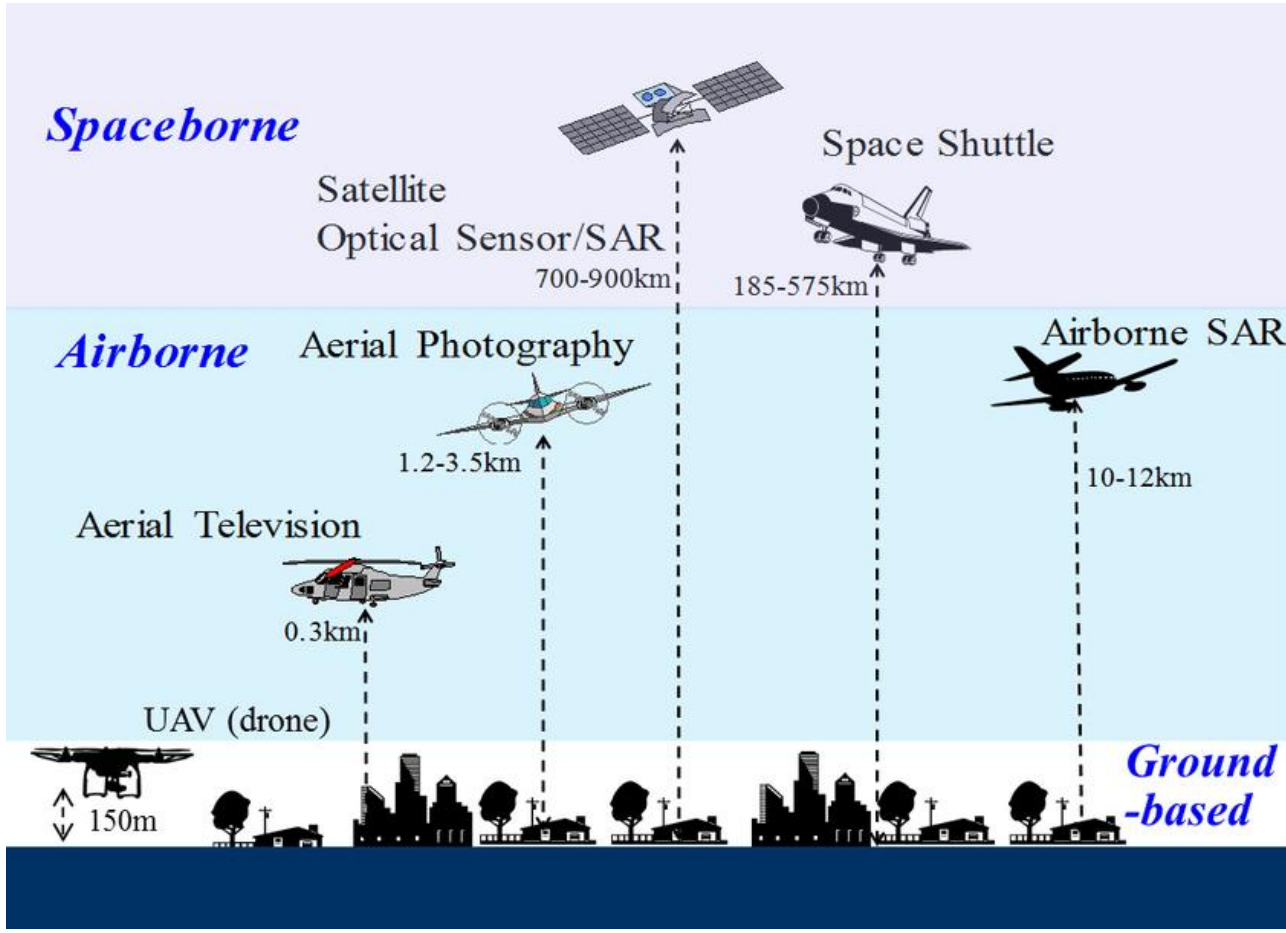


HUMANS AND MACHINES

Sensors



Environmental IoT sensors



Sensors used for remote sensing

HUMANS AND MACHINES
Smart Labels

- “Smart labels” enabling sensing and tracking
- Ynvisible

HUMANS AND MACHINES
Robots

Eco-Robotix

Efficient pesticide application

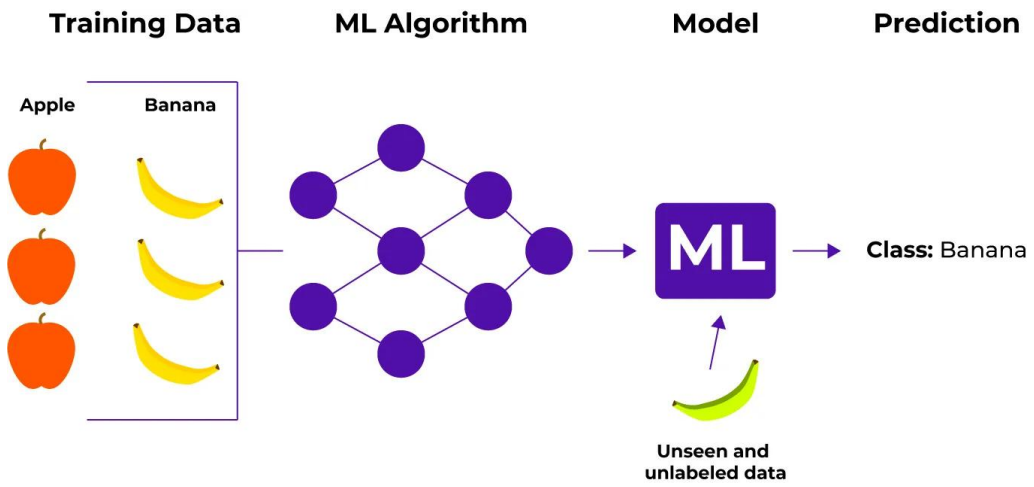
Naio Technologies

Weed elimination

HUMANS, MACHINES AND NATURE

AI fundamentals

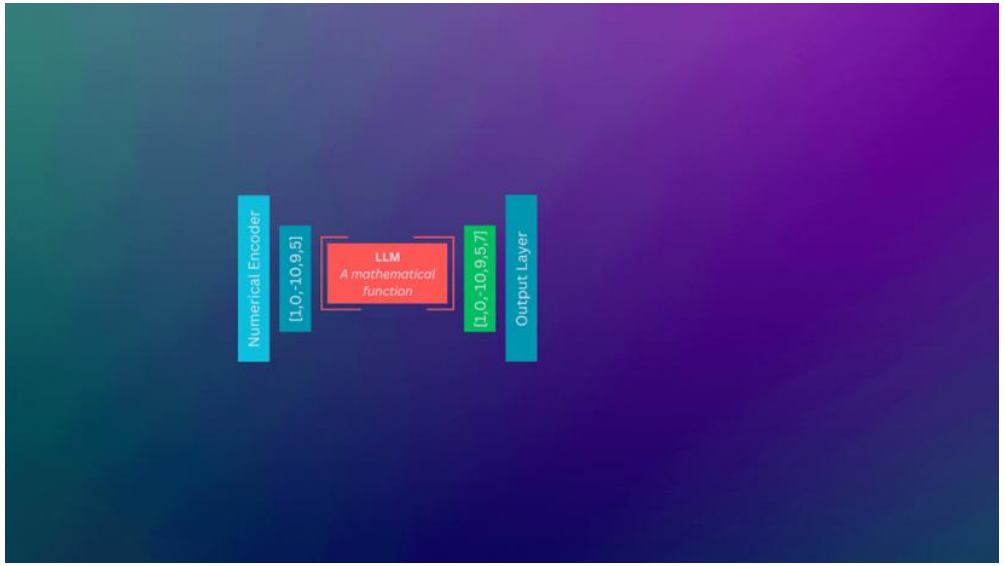
Biological inspired beat symbolic inspired AI- neural networks



An AI Engineer's Guide to Machine Learning and Generative AI

Generative AI

Large Language Models (LLMs) and Large Multimodal Models (LMMs)



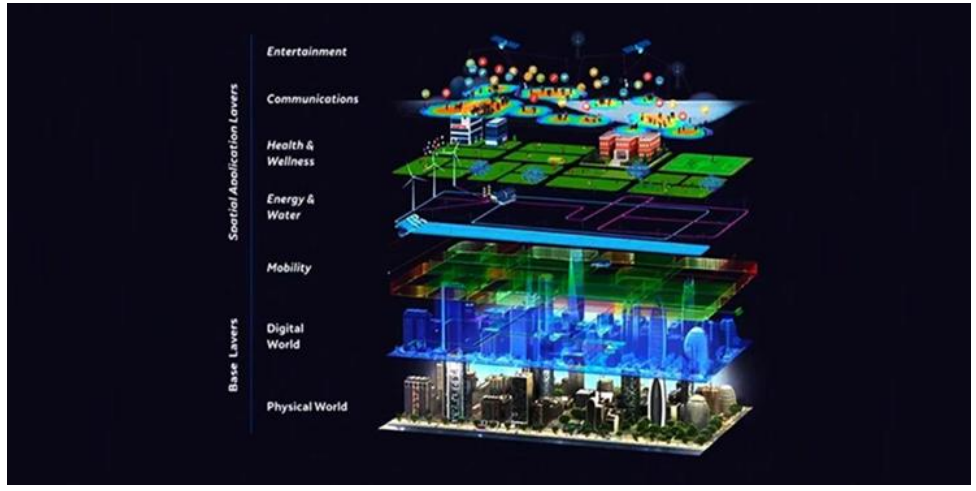
HUMANS, MACHINES AND NATURE

AI/Robotics and Nature

[Large Language Models Empowered Agent-based Modeling and Simulation: A Survey and Perspectives]

AR/VR and Nature

- The Data World
- The Augmented World
- The Digitally Twinned World
- The Virtual World



Plume, combining AI and AR

HUMANS, MACHINES AND NATURE

Nature services

Intrinsic services

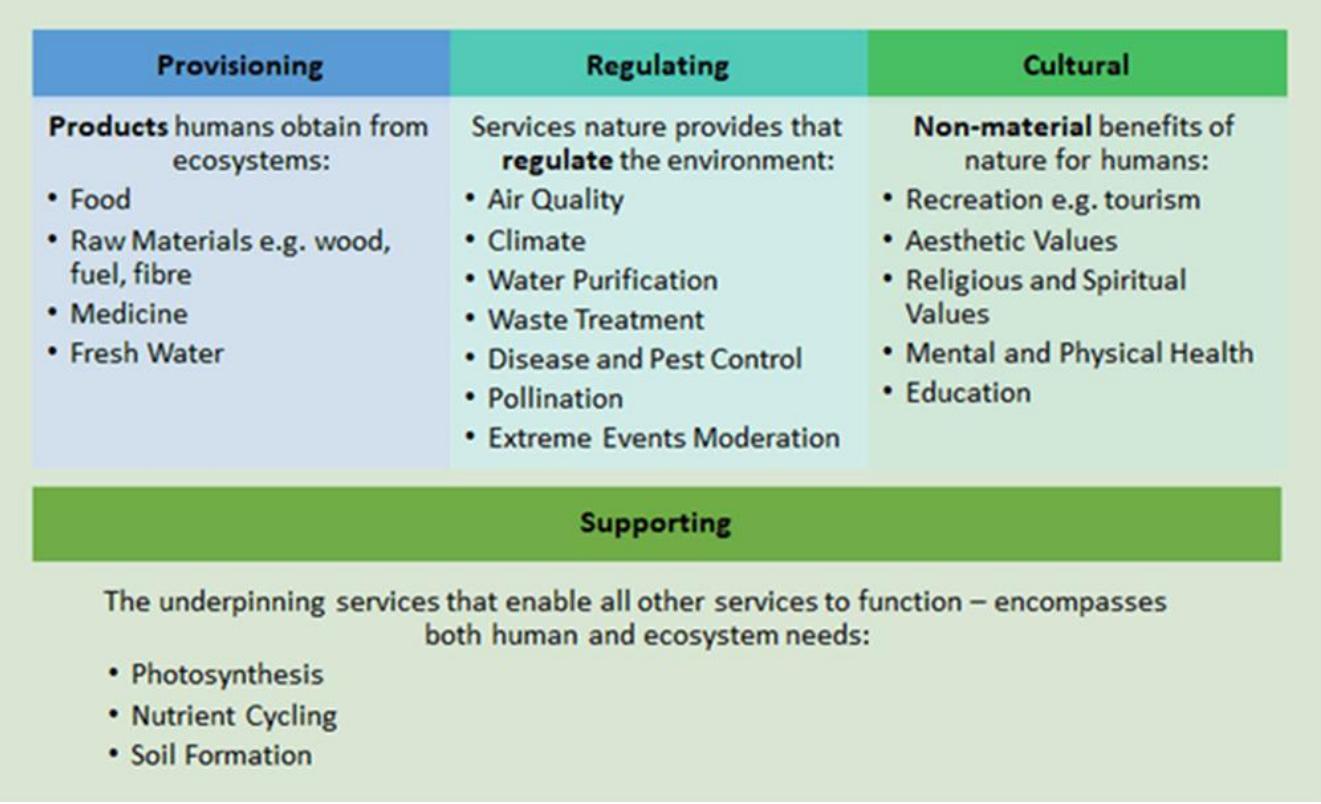
The Real, Data, Augmented and Digitally Twinned Worlds

Offset services

The Digitally Twinned World

Derivative services

The Real and Virtual Worlds



Ecosystem Services: the Fundamentals

INTERNET OF NATURE (ION): a Proposal for Disruption

Internet of Nature (IoN)

Humans and Machines (AI, Robotics, AR/VR) working together will create:

- The Augmented Nature
- The Digital Twinned Nature
- The Virtual Nature

Intrinsic markets will be based on the Augmented Earth

Derivative markets will grow on the Virtual Earth

Offset markets based on the Digital Twinned Earth will start with carbon (forest, soil, blue) and proceed with biodiversity and water markets

Valued at 9, 8 trillion dollars, “Nature” is the largest market in the World

It all started with Al Gore’s Digital Earth

INTERNET OF NATURE (ION): a Proposal for Disruption

Agents will be central*. Examples:

- AA (precision agriculture)
- AB (biodiversity conservation and restoration)
- AC (carbon removal)

Agents will access:

- Data infrastructure- Nature data obtained locally and remotely (including multi-modal and unstructured data, and legacy models)
- Large Multimodal Models (fine tuned with appropriate content reflecting relevant domain knowledge)

*Awesome Agents for Computer Use

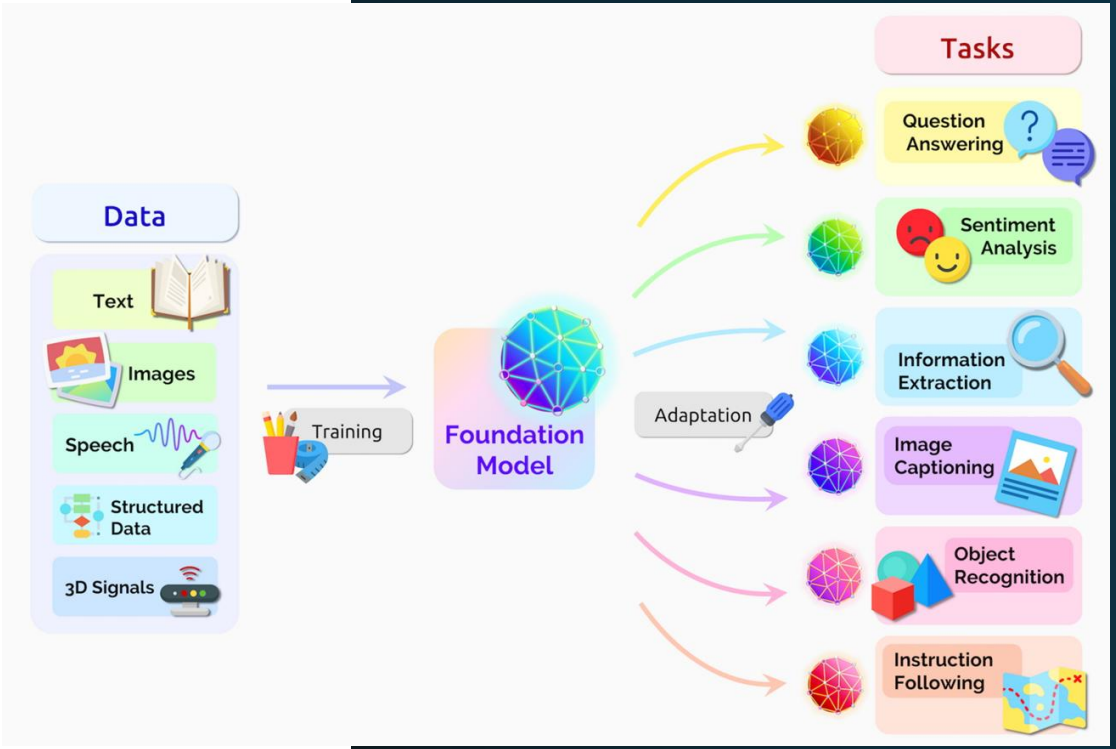
Agents will enable interaction:

Using augmented and virtual representations of Nature

SNAP AR glasses

INTERNET OF NATURE (ION): a Proposal for Disruption

- Develop an IoN Archive, a peer-reviewed journal and yearly conference
- Developing a Nature Foundation Model using attributions to financially reward contributions
- Partnerships with major university groups and companies
- A local World class ecosystem



INTERNET OF NATURE (ION): a Proposal for Disruption

The Community project

A minimalist Craigslist inspired Web project

Resources and tools for everyone

Inspirations

The Handbook of Citizen Science in Ecology and Conservation

The Air Sensor Toolbox

Nature apps

Ask Nature

INTERNET OF NATURE (ION): A Development Plan

Develop prototype Agents via pilot projects (viticulture, precision agriculture, forestry, bio-diversity, water supply management and other areas of interest)- June 2025

Develop a prototype AI Companion for a community project- July 2025

Explora Summit- September 2025

Develop the IoN archive and Nature Foundation Model- 2026

ANTÓNIO CÂMARA

antonio.camara@ydreams.com

Sunday News

The YDreams Collection

Appendix I- Coding a multimodal dynamic model with help from Claude

````
f# Multi-Modal Simulation Model Development Framework

## 1. Data Structure Design

### 1.1 Numerical Variable Handling
```python
class NumericalVariable:
    def __init__(self, value, units, bounds, distribution=None):
        self.value = value
        self.units = units
        self.bounds = bounds
        self.distribution = distribution
        self.history = []

    def update(self, new_value):
        if self.bounds[0] <= new_value <= self.bounds[1]:
            self.history.append(self.value)
            self.value = new_value
```
````

Coding a multimodal dynamic model with help from Claude

```
### 1.2 Linguistic Variable Handling
```python
class LinguisticVariable:
    def __init__(self, value, possible_values, fuzzy_sets=None):
        self.value = value
        self.possible_values = possible_values
        self.fuzzy_sets = fuzzy_sets or {}
        self.history = []

    def update(self, new_value):
        if new_value in self.possible_values:
            self.history.append(self.value)
            self.value = new_value
```
```

Coding a multimodal dynamic model with help from Claude

```
### 1.3 Image Variable Handling
```python
class ImageVariable:
    def __init__(self, image_data, type="static", metadata=None):
        self.image_data = image_data
        self.type = type  # "static" or "dynamic"
        self.metadata = metadata or {}
        self.history = []

    def update(self, new_image_data):
        self.history.append(self.image_data)
        self.image_data = new_image_data
```
```

Coding a multimodal dynamic model with help from Claude

````
## 2. Integration Framework

### 2.1 Variable Relationships
- Define correlation matrices
- Establish causal relationships
- Map dependencies between different variable types
- Define transformation rules

### 2.2 Synchronization Mechanisms
```python
class TimeManager:
    def __init__(self, timestep):
        self.current_time = 0
        self.timestep = timestep
        self.variables = []

    def register_variable(self, variable):
        self.variables.append(variable)

    def advance(self):
        self.current_time += self.timestep
        for var in self.variables:
            var.update_time(self.current_time)
````
````

Coding a multimodal dynamic model with help from Claude

````
## 3. Transformation Rules

### 3.1 Numeric to Linguistic
```python
def numeric_to_linguistic(value, fuzzy_sets):
    memberships = {}
    for set_name, set_func in fuzzy_sets.items():
        memberships[set_name] = set_func(value)
    return max(memberships.items(), key=lambda x: x[1])[0]
```

### 3.2 Linguistic to Numeric
```python
def linguistic_to_numeric(value, fuzzy_sets):
    return fuzzy_sets[value].centroid()
```

### 3.3 Image Processing
```python
def extract_features(image_data):
    features = {
        'numerical': extract_numerical_features(image_data),
        'linguistic': extract_linguistic_features(image_data)
    }
    return features
````
````

Coding a multimodal dynamic model with help from Claude

````
## 4. Simulation Engine

### 4.1 Core Components
```python
class SimulationEngine:
    def __init__(self):
        self.variables = {}
        self.relationships = []
        self.time_manager = TimeManager()

    def add_variable(self, name, variable):
        self.variables[name] = variable

    def add_relationship(self, relationship):
        self.relationships.append(relationship)

    def step(self):
        self.update_variables()
        self.apply_relationships()
        self.time_manager.advance()
```

### 4.2 Update Mechanisms
```python
def update_variables(self):
    for var_name, variable in self.variables.items():
        if variable.needs_update():
            self.update_variable(var_name)
````
````

Coding a multimodal dynamic model with help from Claude

```
## 5. Validation Framework

### 5.1 Numerical Validation
- Statistical tests
- Error metrics
- Confidence intervals

### 5.2 Linguistic Validation
- Semantic consistency checks
- Expert validation
- Fuzzy logic validation

### 5.3 Image Validation
- Feature comparison
- Structural similarity index
- Pattern recognition metrics
```

Coding a multimodal dynamic model with help from Claude

````
## 6. Output Processing

### 6.1 Data Fusion
```python
def fuse_outputs(numerical_data, linguistic_data, image_data):
    fused_output = {
        'numerical': process_numerical(numerical_data),
        'linguistic': process_linguistic(linguistic_data),
        'image': process_image(image_data)
    }
    return fused_output
```

### 6.2 Visualization
```python
def create_visualization(fused_output):
    viz = MultiModalVisualization()
    viz.add_numerical_plot(fused_output['numerical'])
    viz.add_linguistic_summary(fused_output['linguistic'])
    viz.add_image_display(fused_output['image'])
    return viz
```

## 7. Error Handling

### 7.1 Type-Specific Errors
```python
class ValidationError(Exception):
    pass

class TypeMismatchError(Exception):
    pass

class BoundaryError(Exception):
    pass
```
````

Coding a multimodal dynamic model with help from Claude

````
## 7. Error Handling

### 7.1 Type-Specific Errors
```python
class ValidationError(Exception):
    pass

class TypeMismatchError(Exception):
    pass

class BoundaryError(Exception):
    pass
```


### 7.2 Error Recovery
```python
def handle_error(error, variable):
    if isinstance(error, ValidationError):
        return fallback_value(variable)
    elif isinstance(error, TypeMismatchError):
        return convert_type(variable)
    else:
        raise error
```
````

Appendix II- Agentic IoN

A list of potential Agents, with the help of Claude
Prototypes to be developed using MCP

AI/Robotics Applications in Agriculture

1. Autonomous Systems
- Robotic planting and seeding
- Precision crop harvesting
- Autonomous crop monitoring drones
- Weeding and pest control robots
- Autonomous fertilizer application

2. AI-Driven Optimization
- Crop disease detection
- Yield prediction algorithms
- Smart irrigation control
- Crop genetic improvement
- Soil health analysis

AR/VR Applications in Agriculture

1. Training & Visualization
- Crop management simulations
- Equipment operation training
- Soil condition visualization
- Precision agriculture interfaces
- Agricultural technique demonstrations

2. Field Management
- Real-time crop health overlays
- Pest and disease identification
- Irrigation planning visualization
- Yield prediction interfaces
- Remote field monitoring

Agentic IoN

AI and Robotics Applications in Forestry

1. Monitoring Technologies
- Drone-based forest mapping
- AI-driven species identification
- Wildfire early warning systems
- Tree growth and health tracking
- Automated forest inventory

2. Management Solutions
- Robotic tree planting
- Autonomous logging equipment
- Biomass estimation algorithms
- Wildlife corridor monitoring
- Carbon sequestration tracking

AR/VR Applications in Forestry

1. Management Technologies
- Forest terrain mapping
- Tree species identification
- Ecosystem simulation
- Wildlife habitat visualization
- Forest fire risk modeling

2. Conservation Approaches
- Habitat restoration planning
- Species migration tracking
- Biomass estimation
- Carbon sequestration visualization
- Tree growth simulation

Agentic IoN

AI and Robotics Applications in Food Science

1. Production Technologies
- Robotic food processing
- AI quality control systems
- Automated packaging
- Nutritional content analysis
- Flavor profile optimization

2. Safety and Innovation
- Contamination detection
- Predictive food spoilage algorithms
- Robotic fermentation management
- Personalized nutrition AI
- Ingredient interaction modeling

AR/VR Applications in Food Science

1. Production Techniques
- Food processing training
- Quality control visualization
- Nutritional content analysis
- Flavor profile simulation
- Manufacturing process optimization

2. Consumer Interactions
- Product development interfaces
- Cooking technique demonstrations
- Food safety training
- Ingredient interaction modeling
- Sensory experience simulation

Agentic IoN

AI and Robotics Applications in Environmental Sciences

1. Ecosystem Monitoring
- AI-powered biodiversity tracking
- Robotic environmental sampling
- Climate change impact simulation
- Pollution detection systems
- Ecosystem restoration planning

2. Conservation Technologies
- Species migration tracking
- Habitat restoration robots
- AI-driven conservation strategies
- Environmental data collection
- Invasive species management

AR/VR Applications in Environmental Sciences

1. Ecosystem Analysis
- Climate change modeling
- Biodiversity tracking
- Pollution dispersion visualization
- Habitat restoration planning
- Environmental impact simulation

2. Research Tools
- Species migration tracking
- Microclimate visualization
- Environmental sensor data overlay
- Conservation strategy simulation
- Ecological network mapping

Agentic IoN

AI and Robotics Applications in Landscape Architecture

1. Design Innovation
- AI landscape generation
- Robotic terrain modeling
- Microclimate simulation
- Green infrastructure optimization
- Urban ecosystem design

2. Implementation Technologies
- Autonomous landscaping robots
- AI-driven plant selection
- Sustainable design algorithms
- Vegetation growth prediction
- Water resource management

AR/VR Applications in Landscape Architecture

1. Design Technologies
- 3D landscape modeling
- Urban ecosystem simulation
- Green infrastructure planning
- Microclimate visualization
- Site analysis interfaces

2. Implementation Strategies
- Vegetation growth prediction
- Urban design visualization
- Sustainable design optimization
- Terrain modification simulation
- Environmental impact assessment