

<https://doi.org/10.7236/JIIBC.2024.24.3.69>

JIIBC 2024-3-11

수직형 농축수산 팜의 연구 모델 제안

Proposal for Research Model of Agricultural and Fishery Farm Tower

이영수*, 신승중**

Young-Su Lee*, Seung-Jung Shin**

요약 이 논문은 도시 내 지속 가능한 식량 생산을 위한 5층 구조의 수직형 농축수산 팜(팜 타워) 모델을 개발하였다. 해양농장, 축산, 무농약 자동화 농작물 팜을 통합해 자원을 효율적으로 사용하고 환경 영향을 최소화하는 방안을 제안한다. 순환 경제 원칙에 기반한 이 모델은 각 부분의 산출물을 다른 부분의 자원으로 재활용하여 시스템의 효율을 증진시키며 도시 내 유휴 공간을 활용하고, 일자리 창출과 지역 사회 참여를 촉진할 수 있다. 또한, 식량 생산의 탄소 발자국을 줄이고, 식품 안전성을 향상시키는 데 기여할 수 있다.

추가적으로, 이 연구는 진보된 농업 기술이 도시 구조에 통합되어 글로벌 식량 안보 문제에 대응할 수 있는 방안을 모색한다. 이 모델은 기후 변화와 인구 증가로 인한 식량 위기에 대한 잠재적 해결책을 제시하며, 도시 농업의 발전 방향을 제안한다. 향후 연구에서는 실제 구현을 위한 기술적, 정책적 과제를 다루어야 할 것이다.

Abstract This dissertation developed a five-story vertical livestock and fisheries farm (palm tower) model for sustainable food production in cities. It proposes to integrate marine farms, livestock raising, and pesticide-free automated crop farms to efficiently use resources and minimize environmental impact. Based on circular economy principles, the model can recycle the output of each part into resources from the other, increasing the efficiency of the system, utilizing idle space in the city, and promoting job creation and community participation. It can also contribute to reducing the carbon footprint of food production and improving food safety. In addition, the study explores how advanced agricultural technologies can be integrated into urban structures to address global food security challenges. This model presents potential solutions to the food crisis caused by climate change and population growth, and suggests a direction for the development of urban agriculture. Future research should address the technical and policy challenges for practical implementation.

Key Words : Aquaculture, Automated Crop Production, Circular Economy, Environmental Impact, Food Security, Sustainable Agriculture, Urban Agriculture

*정회원, 호서대학교 벤처대학원 융합공학과
**중신회원, 호서대학교 벤처대학원 융합공학과(교신저자)
접수일자 2024년 5월 7일, 수정완료 2024년 5월 27일
게재확정일자 2024년 6월 7일

Received: 7 May, 2024 / Revised: 27 May, 2024 /

Accepted: 7 June, 2024

*Corresponding Author e-mail : expersin@gmail.com

Department of Convergence Engineering, Graduate School of Venture, Hoseo University, Korea

I. 서 론

1. 연구의 필요성

현대 사회는 급속한 인구 증가와 도시화로 인해 식량 안보와 지속 가능한 농업 시스템에 대한 요구가 증가하고 있다. 특히, 기존의 농업 방식은 대규모 토지 사용, 높은 물 소모, 그리고 환경 오염 등 여러 문제를 내포하고 있어, 이를 해결할 수 있는 새로운 방안이 절실히 필요한 시점이며, 이러한 문제를 해결하기 위해 제안된 '수직형 농축수산 팜'은 도시 내에서 지속 가능한 식량 생산을 가능하게 하는 혁신적인 모델로, 농업, 축산, 수산업을 수직적 공간에 통합하여 효율적인 자원 사용과 환경 영향을 최소화하는 것을 목표로 한다^{[1][2]}.

2. 연구 배경 및 범위

이 연구는 수직농업의 설계, 구현, 그리고 환경적, 경제적 타당성을 분석한다. 또한, 다양한 기후 조건과 도시 환경에서의 적용 가능성을 평가한다. 수직형 농축수산 팜은 여러 층으로 구성된 건물에서 식물, 가축, 그리고 수산물 등 동시에 생산하는 시스템이다. 이 시스템은 각 층을 최적화하여 특정 유형의 생산에 맞게 조정되며, 물 순환 시스템과 에너지 효율적인 설계를 통해 지속 가능한 도시 농업을 실현한다. 본 연구는 특히 5층 구조로 설계하여 각 층에서 해양농장, 축산(닭, 돼지, 소), 그리고 농작물 팜을 통합적으로 운영하는 모델을 제안하고자 한다.

해양농장은 해양 생물들을 위한 지속 가능한 양식 환경을 조성하여 바다의 과잉 어획 문제를 완화하고, 수질 오염을 줄이는 데 기여한다. 예를 들어, 해조류를 활용한 시스템은 이산화탄소를 흡수하고, 우수한 수질 관리를 통해 양식 환경을 최적화한다. 이는 해양 생태계의 건강을 증진시키고, 해양 자원의 지속 가능한 이용을 가능하게 한다. 그리고 축산 팜은 폐쇄된 환경에서 질병 관리를 강화하여 축산 동물의 건강을 유지하고, 전염병의 확산을 방지한다. 예를 들어, 공기 정화 시스템과 온도 조절을 통해 축산 동물의 스트레스를 최소화하고, 생산성을 향상시킬 수 있다. 또한, 동물의 배설물을 바이오가스로 전환하여 에너지를 생산함으로써 자원의 재활용을 촉진한다.^{[5][6]}.

2. 수직형 팜의 타당성

축산 팜은 질병을 효과적으로 관리함으로써 동물 복지를 향상시키고, 이는 고기의 품질과 양에 긍정적인 영향을 미친다. 또한, 폐쇄된 환경에서의 축산은 동물 사육 공간을 최소화하면서 생산성을 극대화할 수 있어, 운영 비용을 절감하고, 경제적 수익을 증대시킬 수 있다. 이러한 다양한 기술적 및 생태적 접근 방식을 통합함으로써, 수직형 농축수산 팜은 도시 환경에서의 지속 가능한 식량 생산에 혁신적인 해결책을 제공할 수 있다. 이 모델은 자원의 효율적 사용과 경제적 타당성이라는 두 가지 주요 목표를 달성하는 데 중요한 역할을 할 것이다.

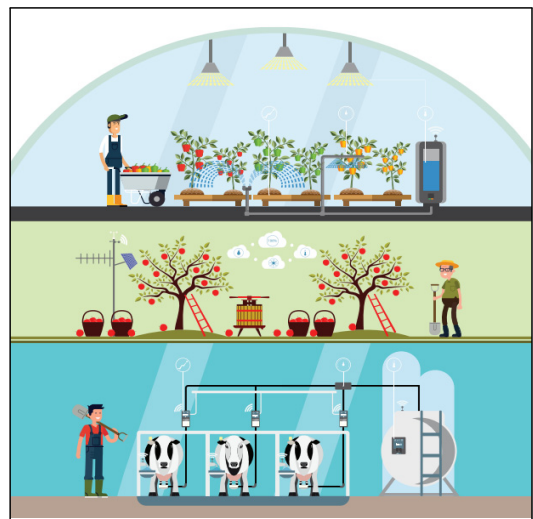
II. 이론적 배경

1. 수직 농업의 발전내용과 현재상황

수직 농업은 제한된 도시 공간에서 지속 가능한 식량 생산을 가능하게 하는 혁신적인 기술이다. 이 장에서는 수직 농업의 기술적 발전을 검토하고, 세계 각지의 도시 농업 적용 사례를 분석하여 이 기술이 어떻게 도시 환경에 통합되고 있는지 설명한다. 특히, 물과 영양분을 순환 사용하는 시스템, LED 조명을 이용한 식물 성장 최적화, 그리고 공간 효율성을 극대화하는 다층 재배 방식 등 현재 수직 농업에서 사용되는 주요 기술들에 대해 다룬다^[3].

III. 수직형 팜의 효율성 및 타당성

1. 수직형 팜의 효율성



출처 : 스마트팜 코리아(www.smartfarmkorea.net)

그림 1. 수직형 팜의 예시

Fig. 1. an example of a vertical farm

IV. 수직형 농축수산 팜 시스템 설계

1. 하드웨어 설계 및 기기

본 연구에서 제안하는 수직형 농축수산 팜은 총 5층 구조로, 각 층은 20평 규모로 설계하였다. 이 구조는 도시 환경에 적합하도록 최적화되어 공간 활용을 극대화. 1층 (해양농장): 수중 환경을 재현하여 다양한 해양 생물을 양식. 최신 수중 LED 조명과 온도 조절 시스템을 설치하여 해양 생물의 성장을 촉진. 2층 (닭 축산): 자동 급여 및 배설물 처리 시스템을 갖춘 축산 공간. 공기 질을 관리하기 위한 고급 환기 시스템을 포함. 3층 (돼지 축산): 각 돼지에 대한 건강 모니터링을 지원하는 스마트 센서와 CCTV를 설치하여 질병 예방과 관리를 자동화. 4층 (소 축산): 대규모 소 사육 공간에 적합한 자동 청소 시스템 및 에너지 효율적인 냉난방 시스템을 설계. 5층 (농작물 팜): 수경재배와 공중 재배를 위한 자동화된 물순환 시스템과 영양 공급 시스템을 구축. LED 성장 램프를 활용하여 광합성을 최적화함.



그림 2. 수직형 농축수산 팜의 구조
 Fig. 2. Structures of Vertical Agricultural and Fisheries Farms

2. 소프트웨어 설계

가. 시스템 아키텍처

소프트웨어 시스템은 파이썬으로 구현되며, 각 층의

환경 제어, 자동화된 공급 체인 관리, 그리고 생산성 모니터링을 위해 통합된 IoT 솔루션으로 구현하였다.

```
# 예시: 기본적인 환경 모니터링 및 제어 시스템

import sensors
import actuators

class FarmTowerControlSystem:
    def __init__(self):
        self.temperature_sensor = sensors.TemperatureSensor()
        self.humidity_sensor = sensors.HumiditySensor()
        self.light_controller = actuators.LightController()
        self.water_pump = actuators.WaterPump()

    def monitor_and_control(self):
        # 온도 및 습도 모니터링
        temperature = self.temperature_sensor.read()
        humidity = self.humidity_sensor.read()

        # 환경 조건에 따른 조명 및 물 공급 제어
        self.light_controller.adjust(temperature, humidity)
        self.water_pump.adjust(temperature, humidity)

        # 로그 데이터 업데이트
        print(f"Temperature: {temperature}C")
```

나. 각 층별 구현

수산 양식 시스템 (1층)의 설계 목적은 어류 및 수생 식물에 적합한 최적의 수질과 온도를 유지하고, 제어 기능으로는 자동 수질 센서와 온도 조절기를 통해 양식 환경을 조절한다. 그리고 축산 시스템 (2층~4층) 설계 목적은 각 층은 닭, 돼지, 소 등 다른 종류의 축산을 위해 맞춤형 환경을 제공하는데 있으며, 제어 기능으로는 공기 질, 온도, 습도를 조절하는 센서와 컨트롤러를 포함하며, 자동 급여 및 배설물 관리 시스템이 중요하다.

그리고 작물 시스템 (5층) 설계 목적은 수경재배 또는 공중재배 같은 방법을 사용하여 고밀도 작물 생산을 지원하고, 제어 기능으로는 조명, 습도, 영양 공급을 자동으로 조절하고, 센서 입력을 바탕으로 조건을 조절하여 식물 성장과 수확량을 최적화하는 데 있다. 다음 구현한 내용은 각층별로 시스템 환경을 고려하여 설계 구현한 내용이다^{[12][13]}.

```
class FarmTower:
    def __init__(self):
        self.floors = {
            1: AquacultureSystem(),
            2: LivestockSystem("chicken"),
            3: LivestockSystem("pig"),
            4: LivestockSystem("cattle"),
            5: CropSystem()
        }
```

```

def operate(self):
    for floor, system in self.floors.items():
        system.monitor_and_control()

class AquacultureSystem:
    def monitor_and_control(self):
        # 수산 양식에 특화된 제어 로직
        print("수산 양식 환경 모니터링 및 제어 중...")

class LivestockSystem:
    def __init__(self, animal_type):
        self.animal_type = animal_type

    def monitor_and_control(self):
        # 동물 유형에 맞는 축산 환경 제어 로직
        print(f"{self.animal_type}의 환경 모니터링 및
제어 중...")

class CropSystem:
    def monitor_and_control(self):
        # 작물 재배에 특화된 제어 로직
        print("작물 재배 환경 모니터링 및 제어 중...")

# 팜 타워 시스템 초기화 및 운영
farm_tower = FarmTower()
farm_tower.operate()

```

V. 블록체인 기술과 NFT 생성

1. NFT 생성기술

NFT 기술을 활용한 프로그램을 작성했고, 우선 각 축산물과 농산물에 대해 디지털 자산을 만들고, 이를 블록체인에 등록하여 각 항목의 고유성과 소유권을 인증하는 접근 방식을 취할 수 있도록 하였다. 예를 들어, 소, 돼지, 닭 등의 축산물과 돔, 광어 같은 어류, 그리고 사과, 배, 포도 같은 농산물 각각을 대상으로 NFT를 생성하고 관리하는 코드를 제시하였다. 다음은 Python을 사용하여 각 항목을 NFT로 변환하고 이를 블록체인에 등록하는 기본적인 내용을 작성 코드이며, 이 코드는 실제 블록체인 네트워크와 연동은 하지 않는 개념을 우선적으로 NFT 생성의 개념적인 구현을 제시하면 다음과 같다^{[8][9]}.

```

from datetime import datetime
import hashlib
import json

class NFT:
    def __init__(self, name, category, details):
        self.name = name
        self.category = category

```

```

        self.details = details
        self.timestamp =
datetime.now().isoformat()
        self.token_id = self.generate_token_id()

    def generate_token_id(self):
        data =
f"{self.name}{self.category}{self.timestamp}"
        return
hashlib.sha256(data.encode()).hexdigest()

    def __repr__(self):
        return json.dumps({
            "name": self.name,
            "category": self.category,
            "token_id": self.token_id,
            "details": self.details,
            "timestamp": self.timestamp
        }, indent=4)

# 각 축산물, 어류 및 농산물 항목에 대해 NFT 생성
livestock = ["소", "돼지", "닭"]
fish = ["돔", "광어"]
crops = ["사과", "배", "포도"]

nft_collection = []

# 축산물 NFT 생성
for animal in livestock:
    nft_collection.append(NFT(name=animal,
category="Livestock", details=f"{animal} NFT"))

# 어류 NFT 생성
for fish_type in fish:
    nft_collection.append(NFT(name=fish_type,
category="Fish", details=f"{fish_type} NFT"))

# 농산물 NFT 생성
for crop in crops:
    nft_collection.append(NFT(name=crop,
category="Crops", details=f"{crop} NFT"))

# 생성된 NFT들 출력
for nft in nft_collection:
    print(nft)

```

2. 블록체인 플랫폼과 통합 및 스마트 컨트랙트 개발

Ethereum 블록체인과의 통합 및 스마트 컨트랙트 개발을 위해서는 Solidity 프로그래밍 언어를 사용하여 스마트 컨트랙트를 작성해보았다. 여기에는 각각의 NFT 생성, 소유권 이전, 판매 등의 기능을 포함할 수 있다. 아래는 Ethereum 블록체인에 NFT를 생성하고 관리하기 위한 간단한 스마트 컨트랙트 예시이다.^{[7][9]}.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import
"@openzeppelin/contracts/token/ERC721/extension/ERC721URIStorage.sol";
import
"@openzeppelin/contracts/access/Ownable.sol";

contract ProductNFT is ERC721URIStorage,
Ownable {
    uint256 private _currentTokenId = 0;

    constructor() ERC721("ProductNFT", "PNFT") {}

    function mintNFT(address recipient, string
memory tokenURI) public onlyOwner returns
(uint256) {
        uint256 newTokenId = _currentTokenId +
1;
        _mint(recipient, newTokenId);
        _setTokenURI(newTokenId, tokenURI);
        _currentTokenId = newTokenId;
        return newTokenId;
    }
}
```

작성한 코드를 설명하면,

가. 컨트랙트 설정

ProductNFT는 ERC721URIStorage와 Ownable을 상속받아, NFT 생성과 URI 설정, 그리고 소유권 관리 기능을 제공한다.

나. 민팅 함수

mintNFT 함수는 NFT를 생성하고, 지정된 주소에게 발행한다. 이 함수는 소유자(컨트랙트 배포자)만 호출할 수 있으며, NFT에 대한 메타데이터 URI를 설정할 수 있다.

다. ERC721 표준

이 컨트랙트는 OpenZeppelin의 ERC721 표준을 사용하여 호환성과 보안을 강화한다.

라. NFT 개발 운영절차 및 필요한 도구

다음의 절차를 통해 실제 블록체인 환경에서 농축수산물의 NFT 프로젝트를 개발하고 운영할 수 있다.

3. NFT 개발 운영절차

가. 스마트 컨트랙트 배포

스마트 컨트랙트를 컴파일하고 Ethereum 네트워크(예: Rinkeby 테스트넷)에 배포한다.

나. 프론트엔드 개발

사용자가 웹 브라우저를 통해 NFT를 손쉽게 민팅하고 거래할 수 있는 인터페이스를 제공하고, 이를 위해 web3.js 또는 ethers.js와 같은 라이브러리를 사용하여 스마트 컨트랙트와 상호 작용할 수 있는 웹 애플리케이션을 개발한다.

다. NFT 메타데이터 관리

NFT에 대한 이미지, 설명 등의 메타데이터를 생성하고 IPFS 같은 분산 파일 시스템에 저장하여 블록체인 외부에서도 데이터를 안전하게 관리할 수 있다.

4. 필요한 도구

가. Node.js

JavaScript 런타임 환경으로, 스마트 컨트랙트 개발과 상호 작용에 필요한 도구들을 실행한다.

나. Truffle Suite

Ethereum 스마트 컨트랙트를 개발, 테스트, 배포하는 데 사용되는 프레임워크이다.

다. Ganache

로컬 블록체인 네트워크를 제공하여 개발 중인 스마트 컨트랙트를 테스트할 수 있게 해준다.

라. Metamask

브라우저 확장 프로그램으로, Ethereum 네트워크와 상호 작용하며 개인 키를 관리한다.

마. Infura

Ethereum 블록체인과 연결할 수 있는 API 서비스를 제공하여, 개발자가 자신의 노드를 운영하지 않고도 네트워크에 접근할 수 있게 한다.

5. 스마트 컨트랙트 배포하기

가. Truffle

프로젝트 설정으로 Truffle을 사용하여 새 프로젝트를 시작한다.

```
npm install -g truffle
truffle init
```

나. 스마트 컨트랙트 작성은 이전에 제공된

ProductNFT 스마트 컨트랙트를 contracts 폴더에 저장한다.

다. 배포 스크립트에 작성하는 것은 migrations 폴더에 배포 스크립트를 작성한다.

```
const ProductNFT =
artifacts.require("ProductNFT");

module.exports = function (deployer) {
  deployer.deploy(ProductNFT);
};

라. truffle-config.js 설정하는 것은 Rinkeby 네트워크에 연결하기 위해 Truffle 설정을 수정한다.

const HDWalletProvider = require('@truffle/hdwallet-provider');
const infuraKey = "YOUR_INFURA_API_KEY";
const mnemonic = "your mnemonic here";

module.exports = {
  networks: {
    rinkeby: {
      provider: () => new HDWalletProvider(mnemonic, `https://rinkeby.infura.io/v3/${infuraKey}`),
      network_id: 4, // Rinkeby의 네트워크 ID
      gas: 5500000, // 가스 한도
    },
  },
  compilers: {
    solc: {
      version: "^0.8.4"
    }
  }
};
```

마. 컨트랙트 배포는 Rinkeby 테스트넷에 컨트랙트를 배포한다.

```
truffle migrate --network rinkeby
```

6. NFT 생성 및 관리

배포된 스마트 컨트랙트를 사용하여 NFT를 생성하고, 이를 웹 애플리케이션을 통해 관리할 수 있으며, 예를 들어, 웹 애플리케이션에서 Metamask를 사용하여 사용자 인증을 받고, mintNFT 함수를 호출하여 NFT를 생성할 수 있다.

이 과정에서 사용자 인터페이스 개발을 위해 React, Vue.js 등의 프레임워크를 사용하고, web3.js 또는 ethers.js 라이브러리를 통해 블록체인과 상호 작용하게 할 수 있다.

7. NFT 생성 및 배포 프로세스

가. 개발 환경 설정

도구: Node.js, Truffle Suite, Ganache, Metamask, Infura

설명: 필요한 도구를 설치하고 개발 환경을 구축한다.

나. 스마트 컨트랙트 개발

도구: Solidity, OpenZeppelin 라이브러리

설명: NFT를 생성하고 관리하기 위한 스마트 컨트랙트를 Solidity 언어로 작성한다.

다. 로컬 테스트

도구: Ganache (로컬 블록체인)

설명: Ganache를 사용하여 스마트 컨트랙트를 로컬 네트워크에서 테스트한다.

라. 배포 스크립트 작성

도구: Truffle Migrations

설명: 스마트 컨트랙트를 블록체인 네트워크에 배포하기 위한 스크립트를 작성한다.

마. 네트워크 설정

도구: Truffle, Infura, Metamask

설명: Truffle 설정 파일(truffle-config.js)에 Rinkeby 테스트넷을 포함하여 네트워크를 구성한다.

바. 블록체인에 배포

도구: Truffle, Infura

설명: Infura를 통해 스마트 컨트랙트를 Rinkeby 테스트넷에 배포한다.

사. NFT 민팅(minting) 및 관리

도구: Metamask, web3.js / ethers.js

설명: 사용자 인터페이스를 통해 스마트 컨트랙트와 상호 작용하며 NFT를 생성(민팅)하고 관리한다.

아. 프론트엔드 개발

도구: HTML/CSS, JavaScript, React/Vue.js

설명: 사용자가 NFT를 손쉽게 민트(minting)하고 관리할 수 있는 웹 애플리케이션을 개발한다.

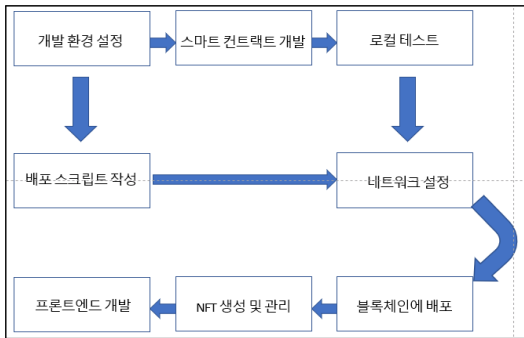


그림 3. NFT 생성 및 배포 프로세스 순서도
 Fig. 3. NFT Creation and Deployment Process Flowchart

V. 결 론

이 논문은 NFT 기술과 팜 타워 모델이 차세대 농업 환경과 디지털 자산 환경에서 혁신적인 해결책으로서 잠재력을 강조하며, 팜 타워에서의 스마트 센서와 같은 기술적 혁신과 NFT에서의 블록체인 기술을 통해 각각의 모델은 자신들의 분야에서 효율성, 생산성, 지속 가능성을 증진할 수 있다. 앞 장에서도 상세히 설명한 NFT 배포 과정은 팜 타워 분석에서 확인된 주요 성공 요인, 즉 기술적 혁신의 적절한 적용과 일치되며, 이는 자동화 및 최적화된 자원 관리와 같은 기술들이 생산성 향상의 핵심으로 작용하듯, NFT 프레임워크는 스마트 계약 및 블록체인을 사용하여 지방 인구 소멸에 필요한 세계 마련에 커다란 기회로 연구될 수 있으며, 디지털 자산을 안전하고 효율적으로 관리하는 것과 같은 효율성 향상을 목

표로 하는 내용의 논문이며, 디지털 자산의 관리에 관한 다양한 연구에 응용할 수가 있을 것이다.

References

- [1] Despommier, Dickson. "The Vertical Farm: Feeding the World in the 21st Century." New York: Thomas Dunne Books, 2010.
- [2] Caplow, Ted. "Building Integrated Agriculture: Urban Farming in NYC." Technology in Society, 2011.
- [3] Thomaier, Susanne, et al. "Farming in and on Urban Buildings: Present Practices and Specific Preconditions of Urban Agriculture from the Technical, Ecological and Planning Perspectives." Ecological Engineering, 2015.
- [4] Beacham, A., Vickers, L., & Monaghan, J. "Vertical farming: A summary of approaches to growing skywards." The Journal of Horticultural Science and Biotechnology, 2021.
- [5] Sanyé-Mengual, E., Oliver-Solà, J., Montero, J. I., & Rieradevall, J. "Economic feasibility of urban rooftop greenhouses: An energy, water, and food nexus analysis." Resources, Conservation and Recycling, 2022.
- [6] Banerjee, C., & Adenaueer, L. "Up, Up and Away! The Economics of Vertical Farming." Journal of Agricultural Studies, 2021.
- [7] Swan, M. Blockchain: Blueprint for a New Economy. O'Reilly Media. 2015.
- [8] Catlett, C. Smart Cities and Artificial Intelligence: Convergent Systems for Planning, Design, and Operations. Elsevier. 2020
- [9] Raval, S. Decentralized Applications: Harnessing Bitcoin's Blockchain Technology. O'Reilly Media. 2016.
- [10] Rhee In-baum, Ryu Daehyun, and Shin Seung Jung. "Technology of Plant Growth Environment Management System Based on USN." Proceedings of the Korea Information Processing Society Conference, Vol. 17, No. 1, pp. 194-197, 2010.
- [11] Baek Yeong-min, Kim In-ho, Park Jin-hyeong, Park Won-chang, Kong Dong-hwan, Shin Seung-jung, and Ryu Daehyun. "Smart Urban Agriculture Management System Using SNS." Proceedings of the Korea Information Processing Society Conference, Vol. 19, No. 2, pp. 1225-1226, 2012.
- [12] Heo Myeong-hoe, and Shin Seung-jung. "Study on Energy Saving House System Using IoT Technology." Journal of the Korean Internet Broadcasting and Telecommunication Society, Vol. 20, No. 6, pp. 109-113, 2020.

저 자 소 개

이 영 수(정회원)



- 1982년 : 국립서울과학기술대학교 토목공학과 졸업
 - 2002년 : 명지대학교산업대학원 지적 GIS학과 졸업(석사)
 - 2024년 : 호서대학교 벤처대학원 융합공학과 재학(박사)
 - 2012 ~ 2017 : (전)대구과학대학교 측지정보과 교수
- 측량 및 지형공간정보 기술사
 - 지적기술사
 - 주관심분야 : GIS, 스마트팜, 드론정보공학

신 승 중(종신회원)



- 1988년 : 세종대학교 대학원 경영학과 졸업(석사)
 - 1994년 : 건국대학교 대학원 전자계산학과 졸업(석사)
 - 2000년 : 국민대학교 대학원 정보관리학과 졸업(박사)
 - 1995년 ~ 2003년 : 중부대학교 정보보호학과 교수
 - 2003년 ~ 2022년 8월 : 한세대학교 ICT융합학과 교수
 - 2022년 9월 ~ 현재 : 호서대학교 벤처대학원 융합공학과 교수
- 주관심분야 : 융합공학, 정보보호, 이동통신, 6차산업