

1. 과제 계획서

☐ 과제 목적 및 필요성

■ 개요

- . ESP32 아두이노 드론을 조립하고, 아두이노 소프트웨어를 이용하여 드론의 딥러닝 학습 및 작동을 수행하고자 함.
- . ESP32 아두이노 드론을 동작하기 위한 아두이노 소프트웨어 설치 및 개발환경 구성

■ 필요성

- . 드론은 탑승 조종사없이 무선전파로 비행 및 조종할 수 있는 무인항공기로, 현재는 고공 촬영, 배달, 농약 살포, 공기 질을 측정, 인공지능, GPS 빅데이터, 광학탐지 등 다양한 기술과 결합하여 다방면에 사용되고 있음.
- . 따라서, 본 프로젝트에서는 ESP32 아두이노 드론을 통해 드론의 조립 및 소프트웨어 구성을 이해하고, ESP32 아두이노 보드에 컴퓨터와 연결하여 아두이노 소프트웨어를 활용하여 ESP 아두이노 드론의 구성 작동 및 딥러닝을 활용하여 모터를 작동시켜 날려보고자 함.

☐ 과제 수행내용

■ 수행내용

- . ESP32 아두이노 드론 조립,
- . ESP32 아두이노 드론을 작동시키기 위한 아두이노 소프트웨어 설치
- . ESP32 아두이노 드론과 컴퓨터와 연결하여 아두이노 소프트웨어를 통한 ESP32 아두이노 드론 작동 실행.

■ 아두이노 드론

- . 아두이노 드론이란 아두이노 보드나 아두이노 호환 보드로 제어하는 드론을 말함. 아두이노 기반의 드론은 많은 DIY maker 들에 의해 만들어짐. 그중 아두파일럿(ardupilot)과 멀티위(multiwii) 두 사이트가 대표적임.
- . 아두파일럿(ardupilot) : 공개 소프트웨어로 드론 외에도 비행기, RC카, 미션 플래너 용으로 사용할 수 있음. ardupilot은 리눅스 재단에서 설립한 드론 코드 프로젝트로도 통합 됨.
- . 멀티위(multiwii) : 공개 소프트웨어로 Wii 모션 플러스 확장과 아두이노 프로 미니 보드를 기반으로 탄생되었음. multiwii는 몇가지 RC 드론 모델을 제어할 수 있고, 여러 가지 센서를 사용할 수 있는 공개 소스임.

■ 과제 구성 및 방법

① ESP32 아두이노 드론 키트를 이용해 조립

- 드론 쉴드 고정판 조립 및, 드론 플라스틱 몸체 결합 후 쉴드 장착
- MPU6050 센서 장착, ESP32 아두이노 장착
- 드론 날개, 몸체 연결, 몸체에 배터리 장착

② ESP32 아두이노 드론 개발 환경 구성

- 아두이노 드론 소프트웨어 설치
- ESP32 아두이노 드론에 컴퓨터 연결하기
- ESP32 아두이노 보드 패키지를 설치
- ESP32 아두이노 드론이 작동할 수 있게 아두이노 보드 패키지를 ESP32로 설정

③ 아두이노 스케치 구조 이해

- 컴퓨터의 아두이노 스케치를 통해 아두이노 보드에 연결함
- 컴파일을 통해 코드의 오류를 확인
- 업로드를 통해 아두이노 보드에 업로드를 실행.
- 시리얼 모니터를 통해 실시간으로 진행되는 상황 확인
- 시리얼 모니터의 통신속도와 보드레이트의 통신속도가 맞아야 실행되는 상황이 보임

④ 아두이노 센서 작동

- digitalWrite 함수를 통해 보드에 달려있는 LED를 제어
- 다른 함수들을 통하여 LED의 딜레이, 밝기 조절이 가능

⑤ 아두이노 모터 작동

- ESP32 드론 모터를 작동
- 모터는 속도 조절, 실행 정지 간격 조절
- MPU6050 모듈을 통해 가속도 자이로 센서를 이용하여 드론의 기울어진 정도와 회전속도를 알려줌

⑥ 아두이노 띄우기

- MPU6050 모듈이 기울기와 회전속도를 조절할 수 있게 딥러닝을 이용
- 드론 모터 작동 및 딥러닝 코드를 이용하여 드론을 띄우기 실습

□ 기대효과 및 활용방안

■ 활용방안

- . 드론 자율주행 분야에서 더 나은 운용 방식을 제공할 수 있음.
- . 엔터테인먼트 분야에서 드론을 활용한 공연을 진행함으로써, 코로나19와 같은 시기에 직접적인 공연장 방문 없이도 드론을 활용하여 관람하거나, 무대 보조와 같은 역할이 가능함.
- . 또한, 인공지능과 드론은 어느 분야에서든 제한 없이 사용이 가능하고, 현재도 농업, 과학, 스포츠, 게임 등 여러 분야에서 사용되고 있는 만큼, 그 분야에 적절한 더 발전한 기술을 제공할 수 있음.

■ 기대효과

- . GPS가 탑재된 드론의 경우, GPS 신호가 원활하지 않은 상황이 생기게 될 수 있는데 더 나은 인공지능기술을 이용하여 GPS를 사용하지 않고도 드론이 장애물을 피해 자율적으로 주행할 수 있는 기술

을 사용할 수 있음.

- . 인공지능 기술과 드론 기술이 점점 발전하면 경제적으로도 큰 영향을 미칠 수 있음.
- . 인공지능과 드론의 기술로 인해 생산성이 높아지면 새로운 제조 및 기술 개발로 인해 이와 관련된 새로운 일자리 창출을 기대할 수 있음.
- . 드론으로 인한 더 빠른 운송으로 인해 시장 및 상품 접근성을 확대할 수 있음.

2. 과제 추진방법

가. 팀원별 역할 분담

연번	성명	담당	수행역할	비고
1	배채정	과제관리	과제관리 및 보고서 작성	팀장
2	김민기	자료수집	자료 수집 및 보고서 작성	팀원
3	박수빈	과제계획	과제계획 및 수행	팀원
4	한승희	자료수집	자료수집 및 보고서 작성	팀원
5				
6				
7				

※ 팀 구성원과 세부적인 역할 분담내용을 기술하고 지역사회 및 기업체 연계과제일 경우 참여기업의 역할을 구체적으로 기재

나. 과제추진 일정

추진 내용	수행기간(월)				비고
	10월	11월	12월	월	
Brain stoming					
자료수집					
실험 및 보고서 작성					

※ 세부추진 내용 및 추진 시기는 구체적으로 기재

다. 참여기관의 제안 내용

없음



캡스톤 디자인 과제 수행 결과보고서



1. 과제수행 및 결과보고

작 품 명	인공지능을 활용한 드론 설계 및 제어
<input type="checkbox"/> 작품 개요 및 필요성	
<div> <div>■ 작품 개요</div> <ul style="list-style-type: none"> . ESP32 아두이노 드론을 조립하고, 아두이노 소프트웨어 환경에서 딥러닝 학습을 통해 드론을 제어함 . ESP32 아두이노 드론의 동작에 필요한 환경 구성을 함 . DNN 인공 신경망 라이브러리를 사용해 드론을 학습시키고 제어함 </div>	
<div> <div>■ 필요성</div> <ul style="list-style-type: none"> . 드론은 드론 경찰, 물류 관리, 시설 관리, 촬영 감독, 산림 관리, 해안선 관리, 드론 스포츠 등 다양한 분야와 접목하여 활용 범위를 넓여가고 있음 . 다양한 분야에서 활용되는 드론이 인공지능과의 만남으로 더욱 정교하게 조종 가능해지면서 각 분야에서의 활약을 기대할 수 있음 . 인공지능을 활용한 드론의 제어 기술의 터득으로 안전사고의 방지를 기대할 수 있음 </div>	
<input type="checkbox"/> 작품의 개발 방법 및 과제 수행 과정	
<div> <div>■ 개발 방법</div> <ul style="list-style-type: none"> . 아두이노 소프트웨어 프로그램을 통해 아두이노 스케치 작성 및 컴파일 후 아두이노 드론 보드에 업로드하여 시리얼 모니터를 통한 결과 확인 . 아두이노 소프트웨어 환경에서 DNN 인공 신경망 라이브러리를 사용해 드론을 학습시키고 제어함 </div>	
<div> <div>■ 수행 과정</div> <div>① ESP32 아두이노 드론 키트를 이용해 조립</div> <ul style="list-style-type: none"> - 드론 쉴드 고정판 조립 및, 드론 플라스틱 몸체 결합 후 쉴드 장착 - MPU6050 센서 장착, ESP32 아두이노 장착 - 드론 날개, 몸체 연결, 몸체에 배터리 장착 <div>② ESP32 아두이노 드론 개발 환경 구성</div> <ul style="list-style-type: none"> - 아두이노 소프트웨어 설치 - 아두이노 보드에 컴퓨터 연결 - ESP32 아두이노 드론에 컴퓨터 연결하기 - ESP32 아두이노 보드 패키지를 설치 - ESP32 아두이노 드론이 작동할 수 있게 아두이노 보드 패키지를 ESP32로 설정 </div>	

③ 아두이노 스케치 구조 이해

- 컴퓨터의 아두이노 스케치를 통해 아두이노 보드에 연결함
- 컴파일을 통해 코드의 오류를 확인
- 업로드를 통해 아두이노 보드에 업로드를 실행.
- 시리얼 모니터를 통해 실시간으로 진행되는 상황 확인
- 시리얼 모니터의 통신속도와 보드레이트의 통신속도가 맞아야 실행되는 상황이 보임

④ 아두이노 센서 작동

- digitalWrite 함수를 통해 보드에 달려있는 LED를 제어
- 다른 함수들을 통하여 LED의 딜레이, 밝기 조절이 가능
ex) 255 delay 함수를 통한 LED의 켜고 끄는 간격 조절

```
255 §  
const int LED = 2;  
  
void setup() {  
  pinMode(LED, OUTPUT);  
  
}  
  
void loop() {  
  digitalWrite(LED, HIGH);  
  delay(50);  
  digitalWrite(LED, LOW);  
  delay(50);  
  
}
```

⑤ 아두이노 모터 작동

- ESP32 드론 모터를 작동
- 모터는 속도 조절, 실행 정지 간격 조절
- MPU6050 모듈을 통해 가속도 자이로 센서를 이용하여 드론의 기울어진 정도와 회전속도를 알려줌
ex) 262 ledcWrite 함수로 모터 속도 조절

```
262 §  
const int fan_pin = 19;  
const int fan_channel = 1;  
const int fan_freq = 1;  
const int fan_resolution = 10;  
  
void setup() {  
  ledcAttachPin(fan_pin, fan_channel);  
  ledcSetup(fan_channel, fan_freq, fan_resolution);  
  
  ledcWrite(fan_channel, 100);  
  
  delay(5000);  
  
  ledcWrite(fan_channel, 0);  
  
}  
  
void loop() {  
  
}
```

⑥ 아두이노 띄우기

- MPU6050 모듈이 기울기와 회전속도를 조절할 수 있게 딥러닝을 이용
- 드론 모터 작동 및 딥러닝 코드를 이용하여 드론을 띄우기 실습

< 딥러닝 라이브러리 구현하기 >

1. 2입력 2출력 행렬 계산식 배열과 함수 정리
 - . 순전파 함수 테스트
 - . 가중치 역전파 함수 테스트

. 딥러닝 라이브러리 정리하기(mydnn)

mydnn

```
void forward_Y(double *X, double *W, double *B, double *Y, int M, int N) {
    for(int n=0;n<N;n++) Y[n] = 0;
    for(int m=0;m<M;m++)
        for(int n=0;n<N;n++)
            Y[n] += X[m]*W[m*N+n];
    for(int n=0;n<N;n++) Y[n] += B[n];
}

double calculate_MSE(double *Y, double *T, int N) {
    double E = 0;
    for(int n=0;n<N;n++) E += (Y[n]-T[n])*(Y[n]-T[n])/2;
    return E;
}

void backward_Yb(double *Yb, double *Y, double *T, int N) {
    for(int n=0;n<N;n++) Yb[n] = Y[n] - T[n];
}

void backward_Xb(double *Xb, double *W, double *Yb, int M, int N) {
    for(int m=0;m<M;m++) Xb[m] = 0;
    for(int m=0;m<M;m++)
        for(int n=0;n<N;n++)
            Xb[m] += Yb[n]*W[m*N+n];
}

void backward_Wb(double *X, double *Wb, double *Yb, int M, int N) {
    for(int m=0;m<M;m++)
        for(int n=0;n<N;n++)
            Wb[m*N+n] = X[m]*Yb[n];
}

void backward_Bb(double *Bb, double *Yb, int N) {
    for(int n=0;n<N;n++) Bb[n] = Yb[n];
}

void backward_Bb(double *Bb, double *Yb, int N) {
    for(int n=0;n<N;n++) Bb[n] = Yb[n];
}

void learning_W(double *W, double lr, double *Wb, int M, int N) {
    for(int m=0;m<M;m++)
        for(int n=0;n<N;n++)
            W[m*N+n] -= lr * Wb[m*N+n];
}

void learning_B(double *B, double lr, double *Bb, int N) {
    for(int n=0;n<N;n++) B[n] -= lr * Bb[n];
}

void sigmoid_Y(double *Y, int N) {
    for(int n=0;n<N;n++) Y[n] = 1/(1 + exp(-Y[n]));
}

void sigmoid_Xb(double *Xb, double *X, int M) {
    for(int m=0;m<M;m++) Xb[m] = Xb[m]*X[m]*(1-X[m]);
}

void tanh_Y(double *Y, int N) {
    for(int n=0;n<N;n++) Y[n] = tanh(Y[n]);
}

void tanh_Xb(double *Xb, double *X, int M) {
    for(int m=0;m<M;m++) Xb[m] = Xb[m]*(1+X[m])*(1-X[m]);
}

void relu_Y(double *Y, int N) {
    for(int n=0;n<N;n++) Y[n] = (Y[n]>0?1:0)*Y[n];
}
```

```

void relu_Xb(double *Xb, double *X, int M) {
    for(int m=0;m<M;m++) Xb[m] = Xb[m] * (X[m]>0?1:0) *1;
}

void print(char * s, double *Y, int N) {
    printf("%s\n", s);
    for(int n=0;n<N-1;n++) printf("%.3f ", Y[n]);
    printf("%.3f\n", Y[N-1]);
}

void print(char * s, double *W, int M, int N) {
    printf("%s\n", s);
    for(int m=0;m<M;m++) {
        printf("[", s);
        for(int n=0;n<N-1;n++)
            printf("%.3f ", W[m*N+n]);
        printf("%.3f]\n", W[m*N+N-1]);
    }
    printf("]\n");
}

```

2. 2입력 2출력 인공 신경망 구현하기

. 입력값 X, 가중치 W, 편향 B, 목표값 T

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \end{bmatrix} = X$$

$$\begin{bmatrix} w_1 & w_3 \end{bmatrix} = \begin{bmatrix} 3 & 5 \end{bmatrix} = W$$

$$\begin{bmatrix} w_2 & w_4 \end{bmatrix} = \begin{bmatrix} 4 & 6 \end{bmatrix}$$

$$\begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix} = B$$

$$\begin{bmatrix} t_1 & t_2 \end{bmatrix} = \begin{bmatrix} 27 & -30 \end{bmatrix} = T$$

** 임의의 값을 설정

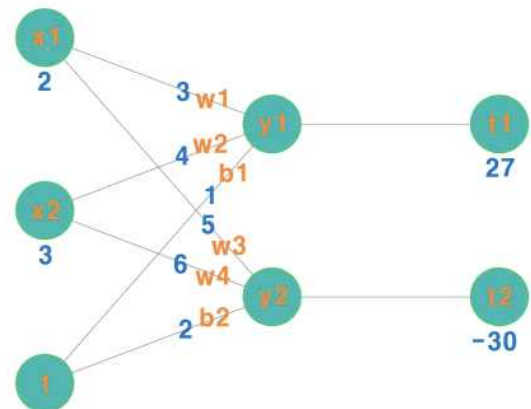
. mydnn의 내용을 추가하여 예제 실행

. 결과

```

Y = [19.000 30.000]
E = 1832.00000000
Yb = [-8.000 60.000]
Xb = [276.000 328.000]
Wb = [
    [-16.000 120.000]
    [-24.000 180.000]
]
Bb = [-8.000 60.000]
W = [
    [3.160 3.800]
    [4.240 4.200]
]
B = [1.080 1.400]

```



3. 3입력 3출력 인공 신경망 구현하기

4. 2입력 2은닉 2출력 인공 신경망 구현하기

5. 가중치 초기화하기

. 정규분포 난수 생성하기 He와 Lecun 가중치 초기화하기

< 딥러닝 라이브러리 활용하기 >

6. 세그먼트 입력 2 진수 출력 인공신경망 입력 데이터 임의로 섞기

7. 은닉층 추가하기

☐ 결과

. 인공지능 기술을 탑재한 드론을 설계하고 직접 제어해 본 결과, LED 불빛의 조작은 물론, 원하는 음을 지정하여 멜로디를 직접 내는 것까지 가능하였음.

. 드론의 날개가 돌아가는 모터의 속도나 위치를 제어할 수 있었음.

. 또한, 반복 횟수를 늘려가며 드론을 제어하는 것과 인공 신경망 구현까지 진행해볼 수 있었음.

. 출력층의 활성화 함수를 소프트맥스, ReLU 등 여러 함수를 사용하여 아두이노에 직접 코드를 작성하고 인공 신경망을 구현함.

2. 활용방안 및 기대효과

☐ 활용방안

. 의학, 기상, 과학, 예술분야 등 다양한 분야에 활용 가능

. 드론을 활용한 인명구조 프로그램 개발 가능

. 드론을 활용하여 시설물 안전진단 가능

. 드론을 활용하여 국토조사 및 순찰 가능

. 드론을 활용하여 벼 쓰러짐 피해 면적 산정에 활용가능

☐ 기대효과

. 농업에 활용하여 시간 단축 및 노동력 절감

. 자연재해로 인해 인명 피해가 발생한 사람이 직접 가기 어려운 지역에서 사망자 수색 가능

. 산림 보호 및 감시 가능

. 원격농장 관리, 정밀농업 확대, 농가당 영농가능 규모 확대 등 농업생산성 향상에 기여

- 범죄현장 매핑과 증거수집, 수색, 정찰 업무, 교통량 모니터링에 빠르게 대처 가능
- 원격 점검을 통해 작업자의 안전 확보

□ 기타

