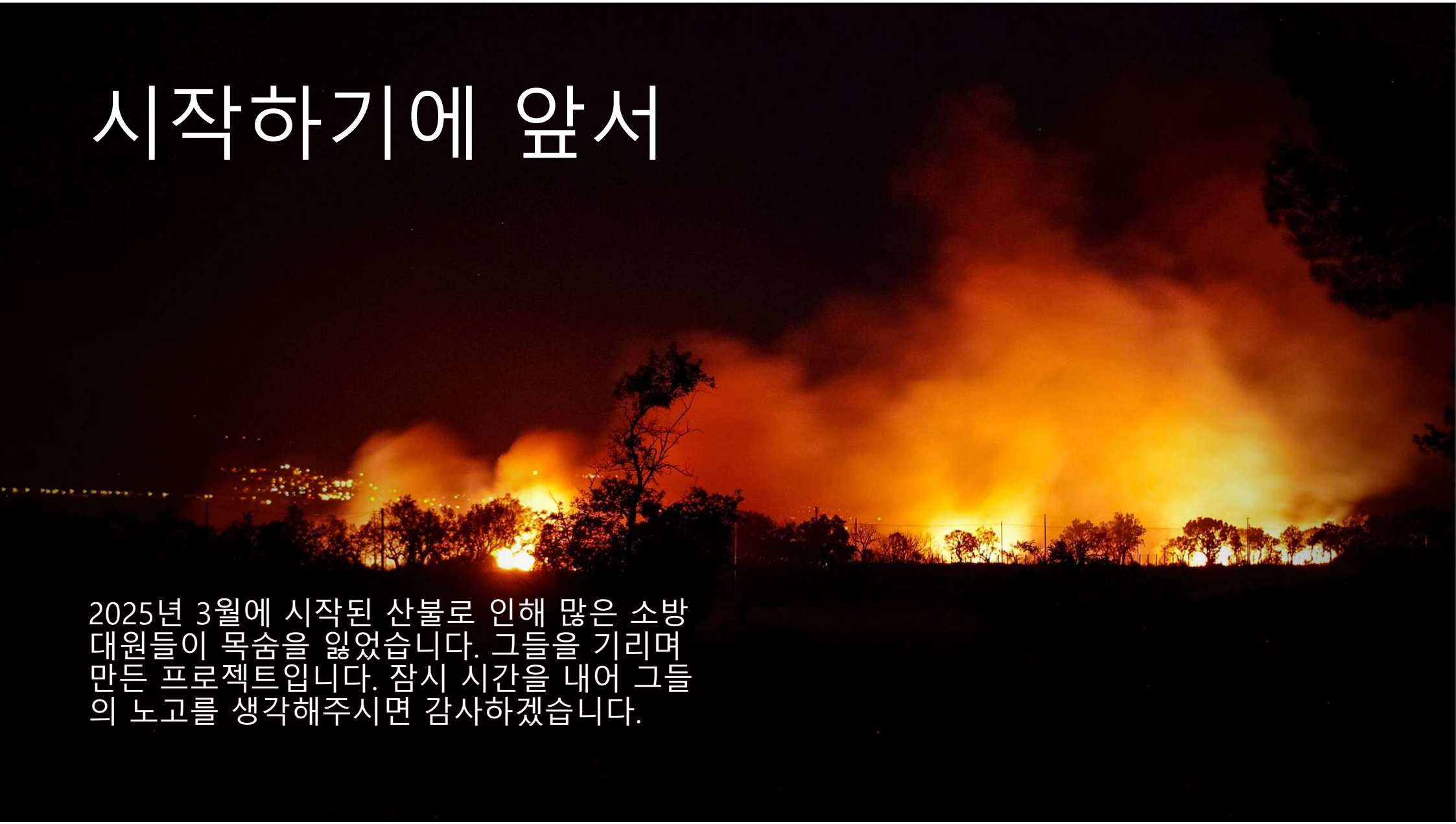
시작하기에 앞서

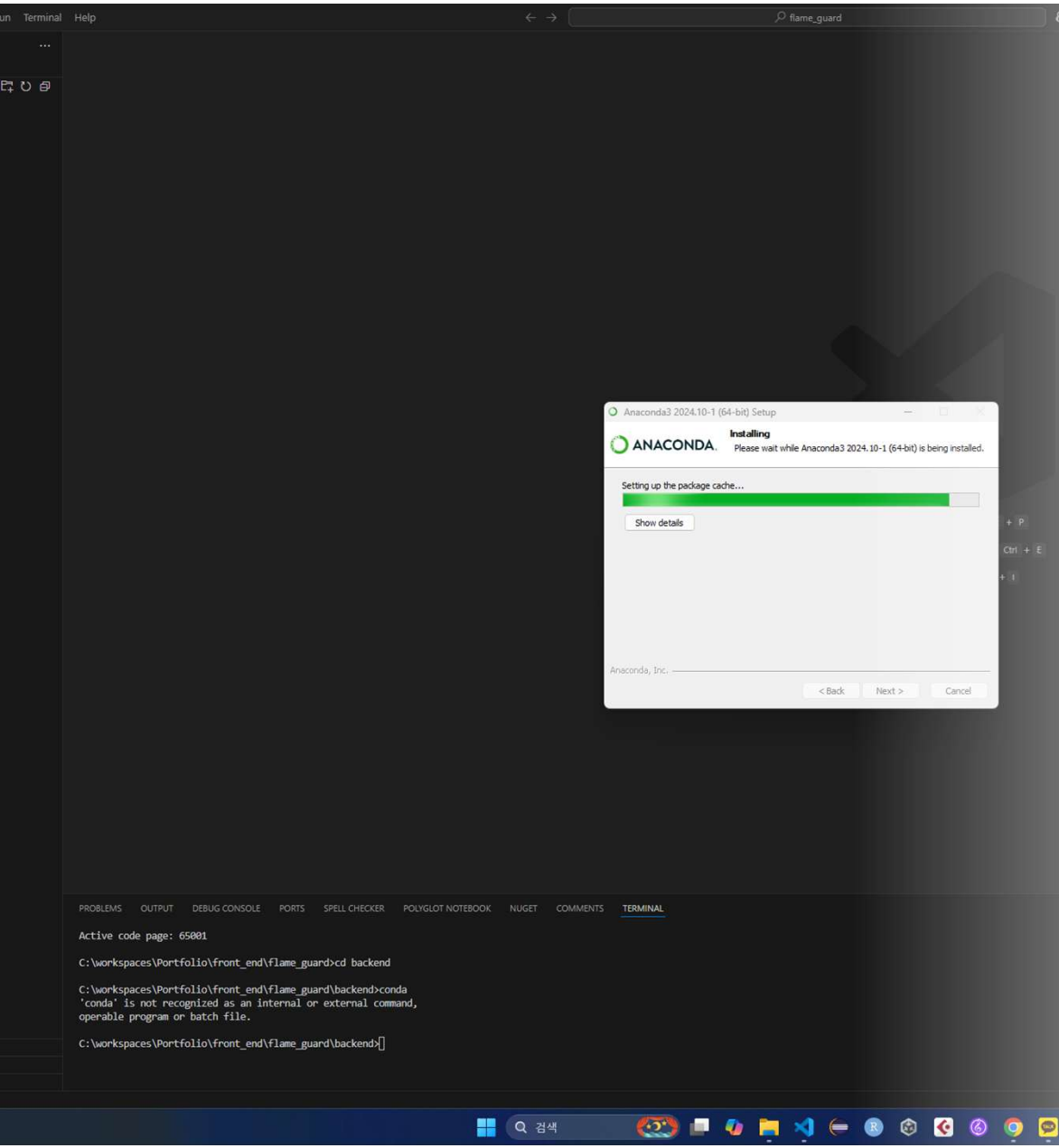


2025년 3월에 시작된 산불로 인해 많은 소방대원들이 목숨을 잃었습니다. 그들을 기리며 만든 프로젝트입니다. 잠시 시간을 내어 그들의 노고를 생각해주시면 감사하겠습니다.

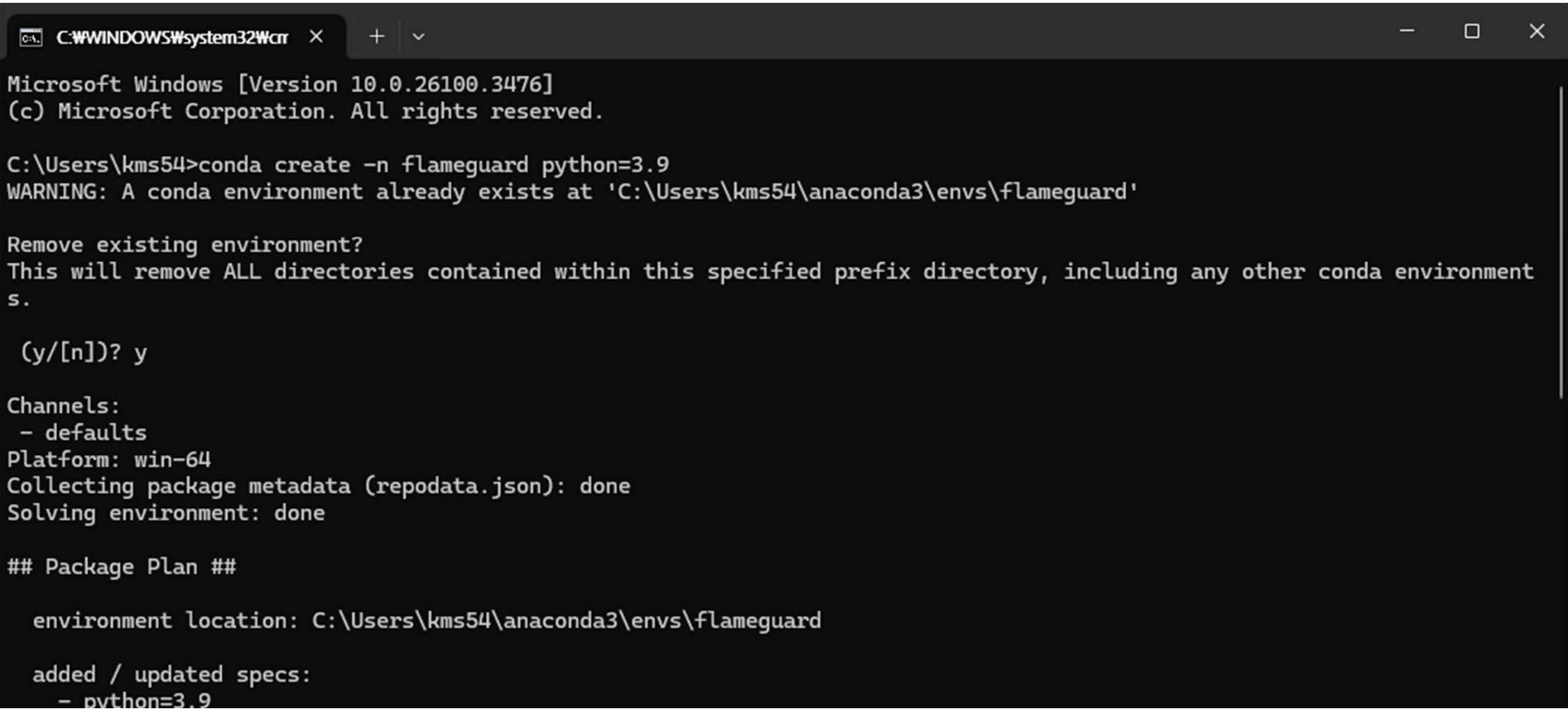
기술 스택

- Next.Js
- Python (Fast API)
- YOLO

개발 환경 구축

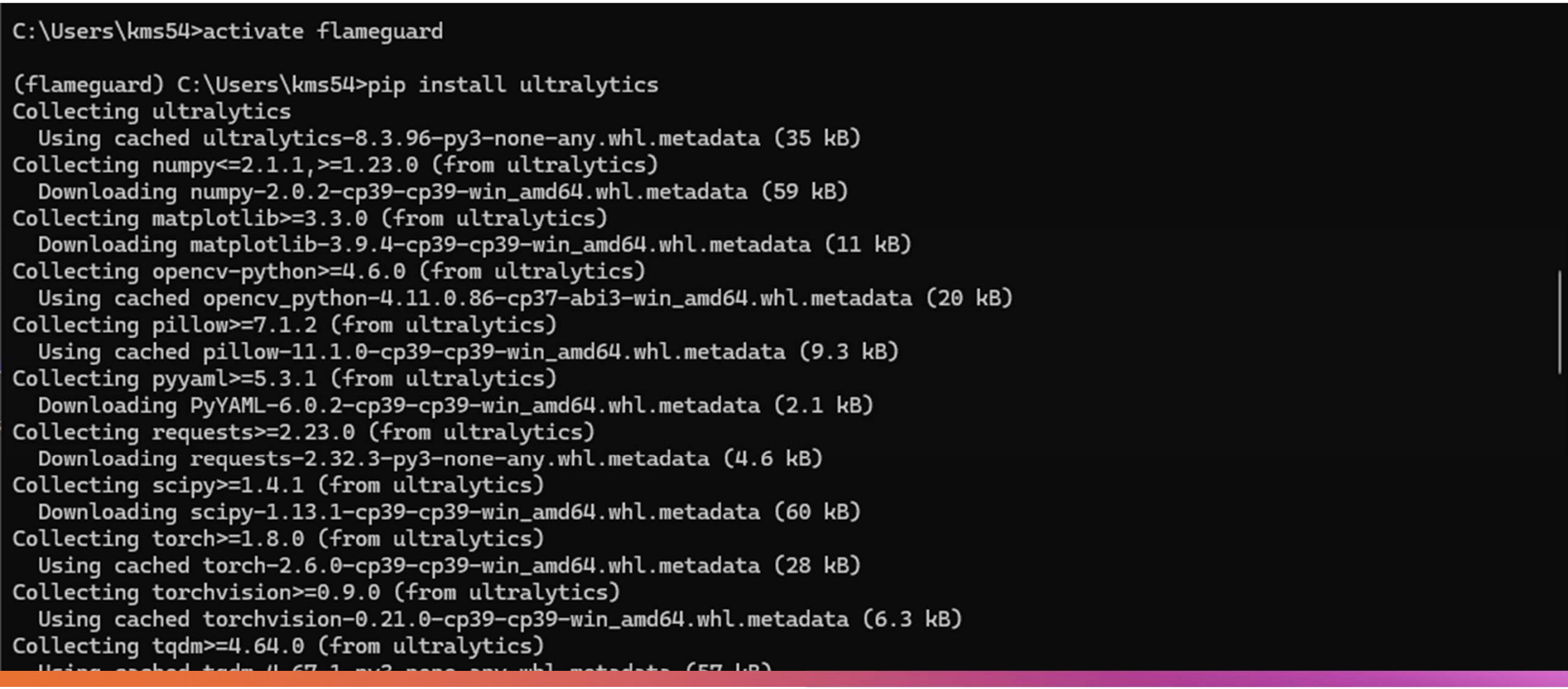


- Git을 통한 기본 틀 다운로드
- Pnpm및 anaconda3 설치

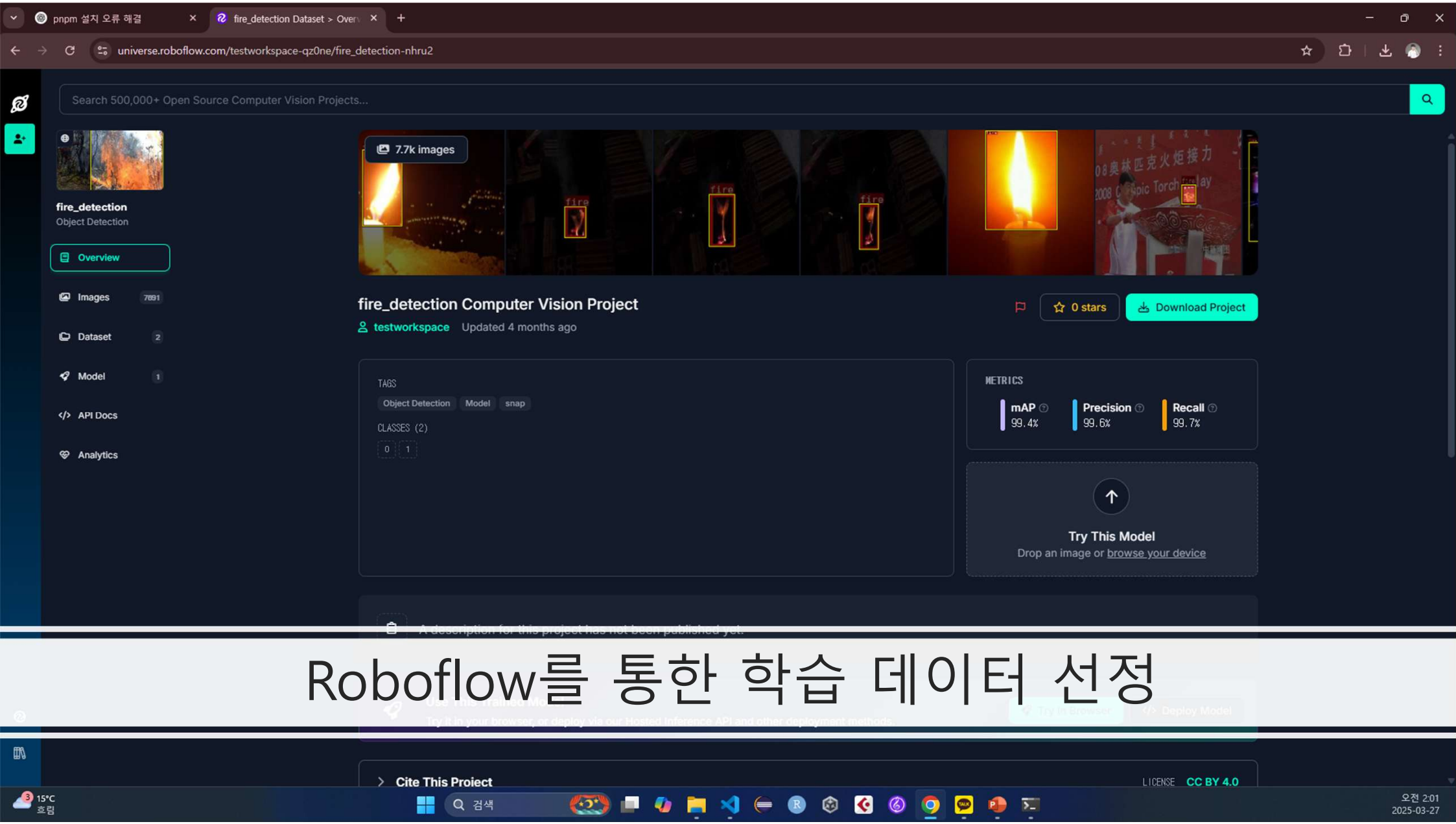


Anaconda를 활용한 환경 구축

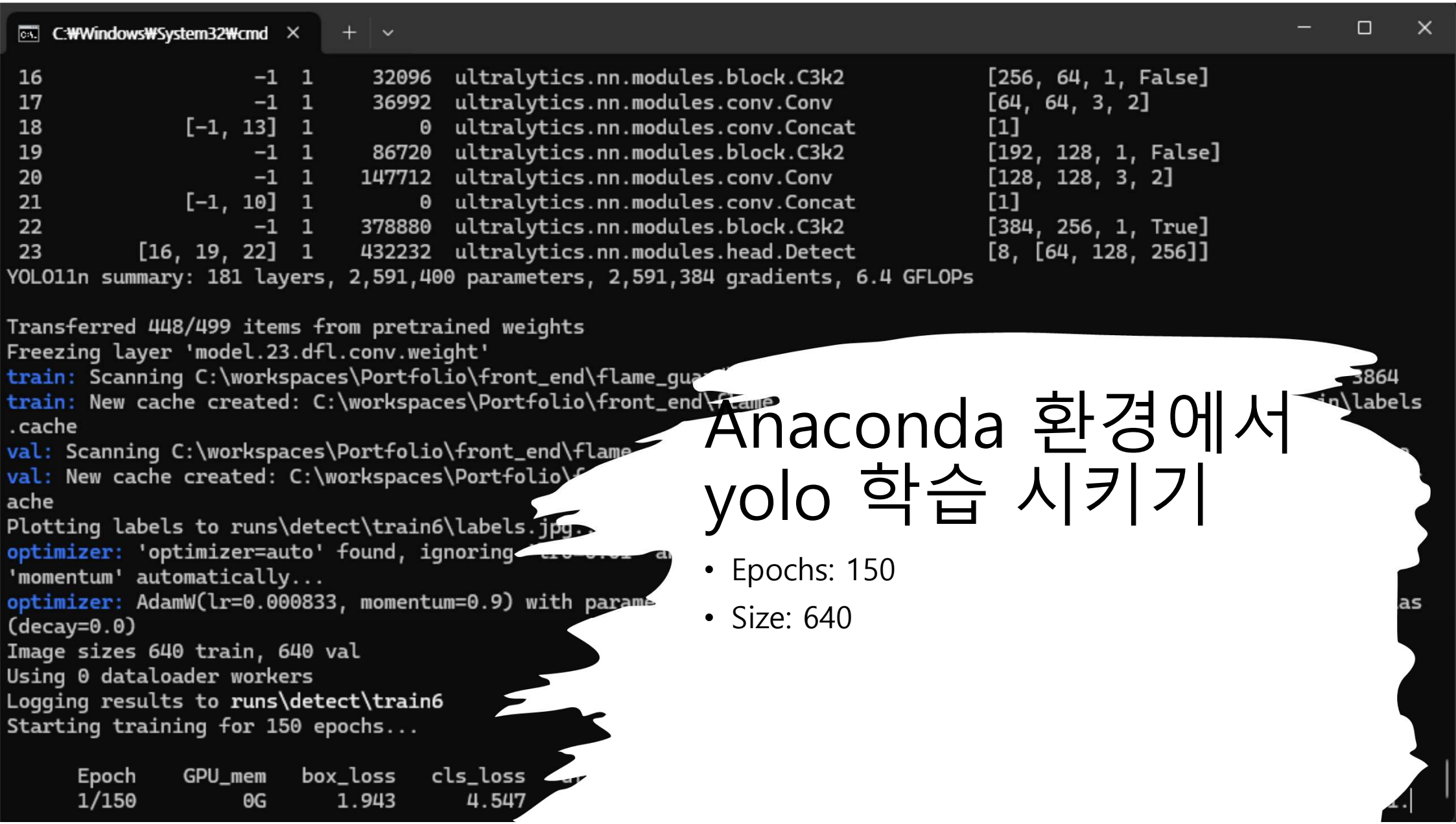
Conda env create –f environment.yml



Ultralytics download

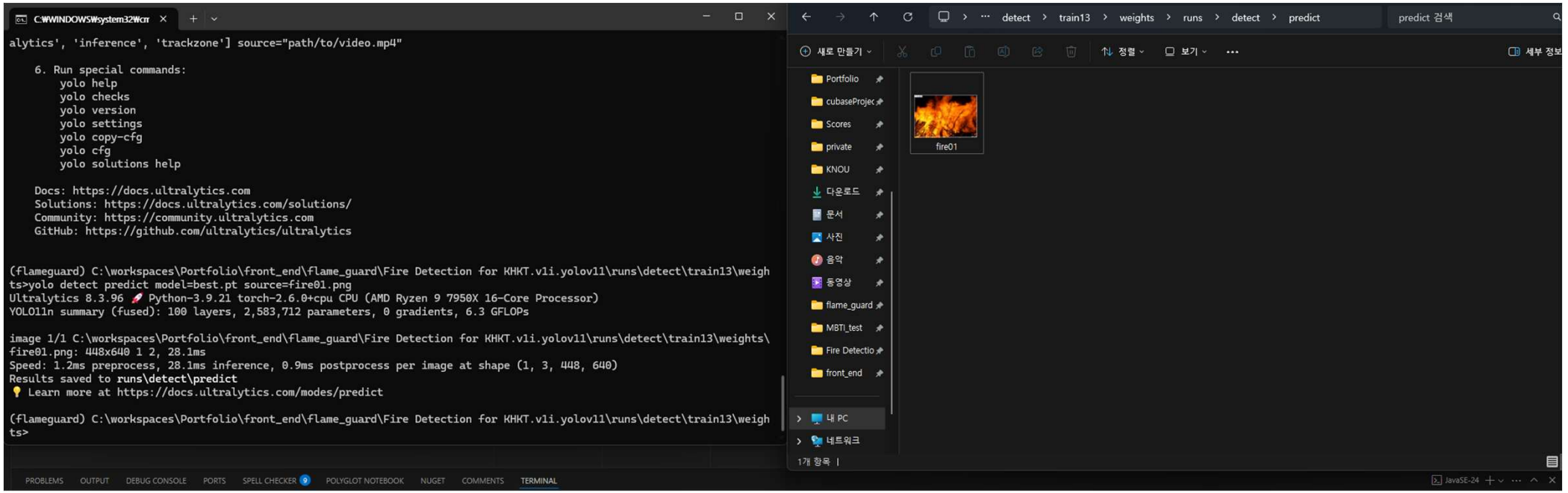


Roboflow를 통한 학습 데이터 선정



Anaconda 환경에서 yolo 학습 시키기

- Epochs: 150
- Size: 640



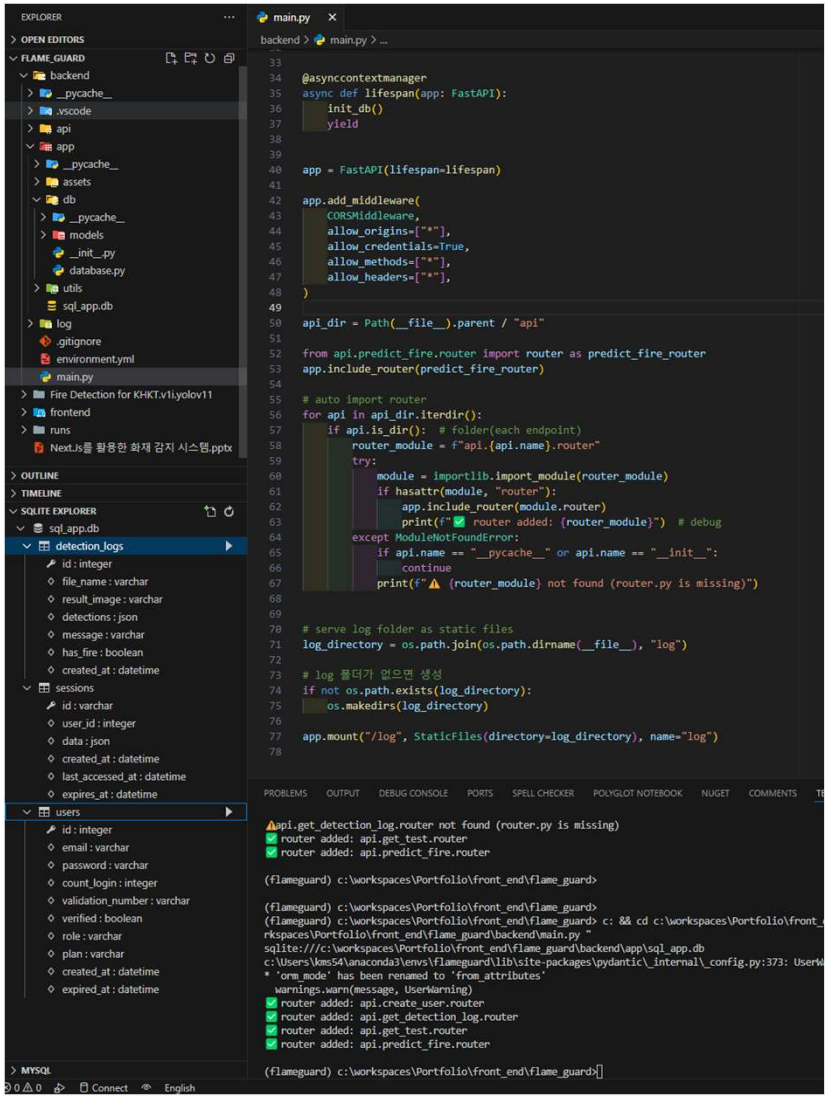
학습된 모델로 결과 도출

Yolo detect predict
model=best.pt
source=fire01.png

화재 감지

다른 색상의 불꽃도 감지 가능

SQLite 연동 성공


EXPLORER
OPEN EDITORS
FLAME_GUARD
backend
__pycache__
.vscode
api
app
__pycache__
assets
db
__pycache__
models
__init__.py
database.py
utils
sql_app.db
log
.gitignore
environment.yml
main.py
Fire Detection for KHKT.v1i.yolov11
frontend
runs
Next.Js를 활용한 화재 감지 시스템.pptx
OUTLINE
TIMELINE
SQLITE EXPLORER
sql_app.db
detection_logs
id : integer
file_name : varchar
result_image : varchar
detections : json
message : varchar
has_fire : boolean
created_at : datetime
sessions
id : varchar
user_id : integer
data : json
created_at : datetime
last_accessed_at : datetime
expires_at : datetime
users
id : integer
email : varchar
password : varchar
count_login : integer
validation_number : varchar
verified : boolean
role : varchar
plan : varchar
created_at : datetime
expired_at : datetime
MYSQL
main.py
backend > main.py > ...
@asynccontextmanager
async def lifespan(app: FastAPI):
init_db()
yield
app = FastAPI(lifespan=lifespan)
app.add_middleware(
CORSMiddleware,
allow_origins=["*"],
allow_credentials=True,
allow_methods=["*"],
allow_headers=["*"],
)
api_dir = Path(__file__).parent / "api"
from api.predict_fire.router import router as predict_fire_router
app.include_router(predict_fire_router)
# auto import router
for api in api_dir.iterdir():
if api.is_dir(): # folder(each endpoint)
router_module = f"api.{api.name}.router"
try:
module = importlib.import_module(router_module)
if hasattr(module, "router"):
app.include_router(module.router)
print(f"✅ router added: {router_module}") # debug
except ModuleNotFoundError:
if api.name == "__pycache__" or api.name == "__init__":
continue
print(f"⚠️ {router_module} not found (router.py is missing)")
# serve log folder as static files
log_directory = os.path.join(os.path.dirname(__file__), "log")
# log 폴더가 없으면 생성
if not os.path.exists(log_directory):
os.makedirs(log_directory)
app.mount("/log", StaticFiles(directory=log_directory), name="log")
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
SPELL CHECKER
POLYGLOT NOTEBOOK
NUGET
COMMENTS
⚠api.get_detection_log.router not found (router.py is missing)
✅ router added: api.get_test.router
✅ router added: api.predict_fire.router
(flameguard) c:\workspaces\Portfolio\front_end\flame_guard>
(flameguard) c:\workspaces\Portfolio\front_end\flame_guard>
(flameguard) c:\workspaces\Portfolio\front_end\flame_guard> c: && cd c:\workspaces\Portfolio\front_
rkspaces\Portfolio\front_end\flame_guard\backend\main.py "
sqlite:///c:\workspaces\Portfolio\front_end\flame_guard\backend\app\sql_app.db
c:\Users\kms54\anaconda3\envs\flameguard\lib\site-packages\pydantic_internal_config.py:373: UserW
* 'orm_mode' has been renamed to 'from_attributes'
warnings.warn(message, UserWarning)
✅ router added: api.create_user.router
✅ router added: api.get_detection_log.router
✅ router added: api.get_test.router
✅ router added: api.predict_fire.router
(flameguard) c:\workspaces\Portfolio\front_end\flame_guard>
Connect
English

구현중입니다

fastAPI