<model_lib_v2_코드 수정 전과 후 비교>

▶ **진행 로그 강화 :** [TOTAL_STEPS], [Step], [LOSS] 콘솔에 실시간 출력과 NaN 감지 추가

▶ **TensorBoard 기록 보강 :** 학습 루프에서 summary_writer로 Loss/total_loss를 직접 기록, 또한 함수 eager_train_step 호출에 summary_writer, global_step 인자 추가

▶ **학습 스텝 산정 로직 추가 :** Epoch.txt를 우선 읽고, 이미지 개수 기반으로 x10, x20(기본값), x30스텝 계산 후 [TOTAL_STEP]출력 (수정 전에는 train_config.num_steps 사용)

▶ **로깅 주기 조건 변경 :** logged_step 차이가 정확히 같을때(==) 에서 이상일 때(>=)로 변경

▶ **체크포인트 복원 경로 수정 :** tf.train.latest_checkpoint(manager.dir)사용으로 워커 경로 문제 대응

▶ **조기종료(early_stop) 정책 개편 :** 고정스텝에서 전체스텝의 70%(early_start)부터 관찰, 종료 시 마지막 체크포인트 저장 후 SystemExit로 종료 (전에는 sys.exit())

▶ **주기적 체크포인트 저장 강화 :** early_start 이후 checkpoint_every_n 간격으로 ckpt_idx 자동저장 및 콘솔 알림

▶ **함수 시그니처 변경 :** train_loop에 fine_tune_checkpoint 파라미터 추가

▶ **모듈 임포트 추가 :** math, absl.logging, functools, distribute_lib, model_lib_v2, shutil, re, logging 추가

▷ **진행/모니터링 가시성 높아짐 :** 스텝, 로스가 매 스텝 출력되고, NaN 값이 뜨면 즉시 중단으로 오류 조기 발견 쉬워짐

▷ **스텝 산정의 자동화/현실화 :** 데이터 개수와 EpochNo,txt를 활용해 상황에 맞게 train_steps 확정

▷ **체크포인트 안정성 :** 복원 시 워커 경로를 고려한 manager_dir 기준으로 최근 체크포인트 탐색

▷ **조기종료 및 저장 정책 개선 :** 학습 70% 이후 성능 정체를 감지해 강제저장 + 안전종료

▷ **TensorBoard 기록 더 촘촘 :** 각 스텝의 loss를 즉시 summary에 기록해 시각화 향상

[NaN값이 생기는 이유]

1. **잘못된 학습 데이터**
   - 라벨이 비어있거나, 값이 비정상적(ex:음수, 무한대 등)일 때
   - 이미지 데이터가 손상되었거나, shape이 맞지 않을 때

2. **모델 구조/하이퍼파라미터 문제**
   - 너무 큰 learning rate
   - 잘못된 초기화(가중치가 너무 크거나 작은 경우)
   - gradient exploding(그레디언트가 너무 커지는 경우)

3. **수치 연산 오류**
   - 0으로 나누기, log(0), sqrt(음수) 등
   - 정규화 과정에서 분모가 0이 되는 경우

4. **Loss 함수 문제**
   - 입력값이 잘못되어 loss 함수 내부에서 NaN이 발생

5. **Mixed Precision 사용 시**
   - bfloat16 등 혼합 정밀도 연산에서 오버플로우/언더플로우

-> tensorflow 버전 이슈로 pip uninstall -y tensorflow-addons
pip install tensorflow-addons==0.12.1 변경

<model_lib_v2_수정.py, model_main_tf2.py, model_main_tf2_FRCNN_res50.py>

▶ annos폴더 내 이미지(.jpg, .jpeg, .png, .bmp) 확장자를 가진 파일만 카운트

▶ 총 스텝 수 = 이미지 개수 * 20

▶ 최대 300,000 STEP, 최소 10,000 STEP 제한

▶ 즉, annos 폴더 내 이미지 파일 개수에 따라 동적으로 train_step을 결정하며,

스텝수는 10,000 ~ 300,000 사이로 제한

▶ 전체 스텝의 70% 이후부터 Early Stopping(조기종료) 조건을 체크

▶ 최적 손실(best_loss)는 0.1미만, patience만큼 loss가 개선되지 않으면 훈련 종료

▶ 70% 이후에는 일정간격(checkpoint_every_n)마다 체크포인트 저장

▶ 환경변수 ENABLE_XLA 값에 따라 TensorFlow의 XLA(JIT) 컴파일러를 활성화 or 비활성화

- XLA(On) = 활성화 : 딥러닝 연산속도 빨라질 수 있음

<Python RadioButton>

▶ 기존과 동일하게 GUI 실행하면 GPU.bat 실행

▶ GUI 실행하면 EpochNo.txt 생성 (annos의 .jpg * 20 값 고정)

▶ RadioButton x10, x20, x30 누르면 그 값에 맞게 곱해져서 자동 저장

▶ 최소 10,000step, 최대 300,000step이고,
그 사이 훈련 step 값은 EpochNo.txt 값으로 고정

▶ 체크포인트 저장은 훈련 스텝 * 0.7 한 값부터 시작