AI + XR : ROADMAP Research, Code, Models, Recs

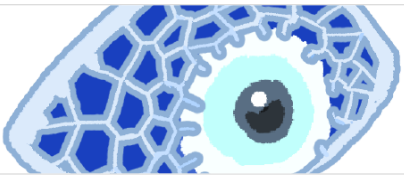
Tags | Code & Resources | DBs & Lists

Latest in AI

camenduru - Overview

camenduru has 1600 repositories available. Follow their code on GitHub.

https://github.com/camenduru



💻 **Non-Profit GPU Cluster**

https://github.com/camenduru/non-profit-gpu-cluster

https://github.com/camenduru/tost-api-examples

🐋 **Docker**

🆕 https://github.com/camenduru/hunyuan-video-runpod

🆕 https://github.com/camenduru/hunyuan-video-docker

🆕 https://github.com/camenduru/ltx-video-docker

🆕 https://github.com/camenduru/wan-docker

🥪 **Tost Apps**

https://github.com/camenduru/memo-tost

https://github.com/camenduru/trellis-tost

https://github.com/camenduru/hunyuan-video-tost

https://github.com/camenduru/sdxl-line-art-style-transfer-tost

https://github.com/camenduru/consisid-tost

https://github.com/camenduru/ominicontrol-tost

https://github.com/camenduru/style-tts-tost

https://github.com/camenduru/flux.1-dev-distill-tost

https://github.com/camenduru/flux.1-dev-redux-tost

https://github.com/camenduru/ltx-video-tost

https://github.com/camenduru/outpaint-flux-fill-tost

https://github.com/camenduru/in-context-lora-tost

https://github.com/camenduru/text-behind-tost

https://github.com/camenduru/instantir-tost

https://github.com/camenduru/dimensionx-tost

https://github.com/camenduru/omnigen-tost

https://github.com/camenduru/idm-vton-tost

https://github.com/camenduru/mochi-1-preview-tost

https://github.com/camenduru/vista-dream-tost

https://github.com/camenduru/stable-diffusion-3.5-large-tost

https://github.com/camenduru/hallo2-tost

https://github.com/camenduru/ichigo-tost

https://github.com/camenduru/inpaint-flux-tost

https://github.com/camenduru/pyramid-flow-i2v-tost

https://github.com/camenduru/pyramid-flow-t2v-tost

https://github.com/camenduru/manga-tost

https://github.com/camenduru/pmrf-tost

https://github.com/camenduru/depth-flow-tost

https://github.com/camenduru/flux.1-dev-upscaler-tost

https://github.com/camenduru/mimic-motion-tost

https://github.com/camenduru/triposr-tost

https://github.com/camenduru/advanced-live-portrait-tost

https://github.com/camenduru/outpaint-flux-tost

https://github.com/camenduru/zest-tost

https://github.com/camenduru/outpaint-tost

https://github.com/camenduru/marigold-depth-normal-tost

https://github.com/camenduru/cogvideox-5b-i2v-tost

https://github.com/camenduru/diff-bir-tost

https://github.com/camenduru/dd-color-tost

https://github.com/camenduru/sam2-remove-video-background-tost

https://github.com/camenduru/pulid-flux.1-tost

https://github.com/camenduru/flux.1-dev-controlnet-depth-tost

https://github.com/camenduru/qwen2-vl-7b-tost

https://github.com/camenduru/train-flux-tost

https://github.com/camenduru/birefnet-tost

https://github.com/camenduru/joy-caption-tost

https://github.com/camenduru/cogvideox-5b-tost

https://github.com/camenduru/onevision-flux.1-dev-tost

https://github.com/camenduru/live-portrait-i2v-tost

https://github.com/camenduru/live-portrait-v2v-tost

https://github.com/camenduru/flux.1-dev-lora-tost

https://github.com/camenduru/flux.1-dev-tost

https://github.com/camenduru/open-sora-plan-1.2-tost

https://github.com/camenduru/comfyui-ultralytics-upscaler-tost

https://github.com/camenduru/kolors-tost

https://github.com/camenduru/lumina-next-compositional-tost

https://github.com/camenduru/tost-tts

https://github.com/camenduru/comfyui-ipadapter-animatediff-tost

https://github.com/camenduru/tiled-controlnet-upscaler-tost

https://github.com/camenduru/style-tts-muse-talk-tost

https://github.com/camenduru/t2v-turbo-tost

https://github.com/camenduru/stable-audio-tost

https://github.com/camenduru/sam2-remove-video-background-tost

https://github.com/camenduru/comfyui-instantid-ipadapter-controlnet-facedetailer-tost

https://github.com/camenduru/comfyui-ipadapter-latentupscale-tost

https://github.com/camenduru/toon-crafter-tost

https://github.com/camenduru/muse-pose-tost

https://github.com/camenduru/dw-pose-tost

https://github.com/camenduru/colorize-line-art-tost

https://github.com/camenduru/meta-llama-3.1-8b-instruct-tost

https://github.com/camenduru/dust3r-tost

https://github.com/camenduru/instant-mesh-tost

https://github.com/camenduru/ic-light-tost

https://github.com/camenduru/magic-time-tost

https://github.com/camenduru/sdxl-cyberrealistic-tost

https://github.com/camenduru/sdxl-turbo-tost

3D ML Papers

https://github.com/camenduru/TRELLIS-jupyter

https://github.com/camenduru/MASt3R-jupyter

https://github.com/camenduru/MeshAnythingV2-jupyter

https://github.com/camenduru/Unique3D-jupyter

https://github.com/camenduru/InstantMesh-jupyter

https://github.com/camenduru/GRM-jupyter

https://github.com/camenduru/GeoWizard-jupyter

https://github.com/camenduru/CRM-jupyter

https://github.com/camenduru/TripoSR-jupyter

https://github.com/camenduru/DSINE-jupyter

https://github.com/camenduru/dust3r-jupyter

https://github.com/camenduru/LGM-jupyter

https://github.com/camenduru/3DTopia-jupyter

https://github.com/camenduru/Depth-Anything-jupyter

https://github.com/camenduru/3DFauna-colab

https://github.com/camenduru/HarmonyView-colab

https://github.com/camenduru/OpenLRM-colab

https://github.com/camenduru/BEV-colab

https://github.com/camenduru/buddi-colab

https://github.com/camenduru/HumanGaussian-colab

https://github.com/camenduru/GeoDream-colab

https://github.com/camenduru/4dfy-colab

https://github.com/camenduru/LucidDreamer-colab

https://github.com/camenduru/Wonder3D-colab

https://github.com/camenduru/zero123plus-colab

https://github.com/camenduru/GaussianDreamer-colab

https://github.com/camenduru/MVDream-colab

https://github.com/camenduru/dreamgaussian-colab

https://github.com/camenduru/Text2Tex-colab

https://github.com/camenduru/SyncDreamer-colab

https://github.com/camenduru/SyncDreamer-docker

https://github.com/camenduru/threestudio-colab

https://github.com/camenduru/IT3D-text-to-3D-colab

https://github.com/camenduru/cotracker-colab

https://github.com/camenduru/ZoeDepth-colab

https://github.com/camenduru/zero123-colab

https://github.com/camenduru/PanoHead-colab

https://github.com/camenduru/bite-colab

https://github.com/camenduru/ECON-colab

https://github.com/camenduru/shap-e-colab

🐓 **3D Motion Papers**

https://github.com/camenduru/StableAnimator-jupyter

https://github.com/camenduru/ExAvatar-jupyter

https://github.com/camenduru/GVHMR-jupyter

https://github.com/camenduru/LivePortrait-jupyter

https://github.com/camenduru/UniAnimate-jupyter

https://github.com/camenduru/MimicMotion-jupyter

https://github.com/camenduru/MuseTalk-jupyter

https://github.com/camenduru/MusePose-jupyter

https://github.com/camenduru/EMAGE-jupyter

https://github.com/camenduru/ScoreHMR-jupyter

https://github.com/camenduru/MagicDance-jupyter

https://github.com/camenduru/havatar-colab

https://github.com/camenduru/PyMAF-X-colab

https://github.com/camenduru/STAF-colab

https://github.com/camenduru/BEAT-colab

https://github.com/camenduru/MotionGPT-colab

https://github.com/camenduru/insactor-colab

https://github.com/camenduru/MoMask-colab

https://github.com/camenduru/FineMoGen-colab

https://github.com/camenduru/SMPLer-X-colab

https://github.com/camenduru/MotionDiffuse-colab

https://github.com/camenduru/NIKI-colab

https://github.com/camenduru/PHALP-colab

https://github.com/camenduru/DWPose-colab

https://github.com/camenduru/4D-Humans-colab

https://github.com/camenduru/vid2avatar-colab

https://github.com/camenduru/PARE-colab

https://github.com/camenduru/VIBE-colab

https://github.com/camenduru/ViTPose-colab

📸 **NeRF + Gaussian Splatting**

https://github.com/camenduru/ges-splatting-jupyter

https://github.com/camenduru/4DGen-colab

https://github.com/camenduru/LucidDreamer-Gaussian-colab

https://github.com/camenduru/PeRF-colab

https://github.com/camenduru/4DGaussians-colab

https://github.com/camenduru/neuralangelo-colab

https://github.com/camenduru/gaussian-splatting-colab

https://github.com/camenduru/instant-ngp-colab

📽️ **Video ML Papers**

https://github.com/camenduru/memo-jupyter

https://github.com/camenduru/hunyuan-video-jupyter

https://github.com/camenduru/ConsisID-jupyter

https://github.com/camenduru/echomimic-jupyter

https://github.com/camenduru/JoyVASA-jupyter

https://github.com/camenduru/TANGO-jupyter

https://github.com/camenduru/CogVideoX-5B-jupyter
https://github.com/camenduru/Video-Infinity-jupyter
https://github.com/camenduru/ExVideo-jupyter
https://github.com/camenduru/Diffutoon-jupyter
https://github.com/camenduru/hallo-jupyter
https://github.com/camenduru/ShareGPT4Video-jupyter
https://github.com/camenduru/t2v-turbo-jupyter
https://github.com/camenduru/V-Express-jupyter
https://github.com/camenduru/ToonCrafter-jupyter
https://github.com/camenduru/ID-Animator-jupyter
https://github.com/camenduru/MagicTime-jupyter
https://github.com/camenduru/Open-Sora-Plan-jupyter
https://github.com/camenduru/AniPortrait-jupyter
https://github.com/camenduru/AnimateDiff-Lightning-jupyter
https://github.com/camenduru/Open-Sora-jupyter
https://github.com/camenduru/Magic-Me-jupyter
https://github.com/camenduru/SketchVideo-jupyter
https://github.com/camenduru/FreeNoise-AnimateDiff-colab
https://github.com/camenduru/dreamtalk-colab
https://github.com/camenduru/MotionCtrl-colab
https://github.com/camenduru/LongAnimateDiff-colab
https://github.com/camenduru/PIA-colab
https://github.com/camenduru/FreeInit-colab
https://github.com/camenduru/dynamiCrafter-colab
https://github.com/camenduru/MotionDirector-colab
https://github.com/camenduru/SEINE-colab
https://github.com/camenduru/LaVie-colab
https://github.com/camenduru/stable-video-diffusion-colab
https://github.com/camenduru/SadTalker-colab
https://github.com/camenduru/Show-1-colab
https://github.com/camenduru/VideoCrafter-colab
https://github.com/camenduru/Hotshot-XL-colab
https://github.com/camenduru/video-retalking-colab
https://github.com/camenduru/ProPainter-colab
https://github.com/camenduru/TokenFlow-colab
https://github.com/camenduru/I2VGen-XL-colab
https://github.com/camenduru/CoDeF-colab
https://github.com/camenduru/AnimateDiff-colab
https://github.com/camenduru/Rerender-colab
https://github.com/camenduru/3d-photo-inpainting-colab

https://github.com/camenduru/text2video-zero-colab

https://github.com/camenduru/text-to-video-synthesis-colab

https://github.com/camenduru/one-shot-talking-face-colab

https://github.com/camenduru/wav2lip-colab

https://github.com/camenduru/pix2pix-video-colab

🎙️ **Audio ML Papers**

🆕 https://github.com/camenduru/ACE-Step-jupyter

https://github.com/camenduru/dia-jupyter

https://github.com/camenduru/csm-1b-jupyter

https://github.com/camenduru/DiffRhythm-jupyter

https://github.com/camenduru/FluxMusic-jupyter

https://github.com/camenduru/FoleyCrafter-jupyter

https://github.com/camenduru/stable-audio-jupyter

https://github.com/camenduru/ChatMusician-jupyter

https://github.com/camenduru/NeMo-ASR-jupyter

https://github.com/camenduru/Image2SoundFX-jupyter

https://github.com/camenduru/metavoice-jupyter

https://github.com/camenduru/MAGNeT-colab

https://github.com/camenduru/resemble-enhance-colab

https://github.com/camenduru/OpenVoice-colab

https://github.com/camenduru/singing-voice-conversion-colab

https://github.com/camenduru/styletts-colab

https://github.com/camenduru/HierSpeech_TTS-colab

https://github.com/camenduru/AudioSep-colab

https://github.com/camenduru/coqui-XTTS-colab

https://github.com/camenduru/VALL-E-X-colab

https://github.com/camenduru/seamless-m4t-colab

https://github.com/camenduru/audiogen-colab

https://github.com/camenduru/LP-Music-Caps-colab

https://github.com/camenduru/vampnet-colab

https://github.com/camenduru/tortoise-tts-colab

https://github.com/camenduru/MusicGen-colab

https://github.com/camenduru/elevenlabs-colab

https://github.com/camenduru/Retrieval-based-Voice-Conversion-WebUI-colab

https://github.com/camenduru/whisper-jax-colab

https://github.com/camenduru/bark-colab

https://github.com/camenduru/audioldm-colab

🧨 **Diffusers**

https://github.com/camenduru/AniDoc-jupyter

https://github.com/camenduru/ColorFlow-jupyter

https://github.com/camenduru/ominicontrol-jupyter

https://github.com/camenduru/text-behind-jupyter

https://github.com/camenduru/Meissonic-jupyter

https://github.com/camenduru/PMRF-jupyter

https://github.com/camenduru/AdvancedLivePortrait-jupyter

https://github.com/camenduru/diffusers-image-outpaint-jupyter

https://github.com/camenduru/diffusion-e2e-ft-jupyter

https://github.com/camenduru/Hyper-FLUX-jupyter

https://github.com/camenduru/PaintsUndo-jupyter

https://github.com/camenduru/Kolors-jupyter

https://github.com/camenduru/AutoStudio-jupyter

https://github.com/camenduru/Lumina-Next-jupyter

https://github.com/camenduru/Omost-jupyter

https://github.com/camenduru/HunyuanDiT-jupyter

https://github.com/camenduru/IC-Light-jupyter

https://github.com/camenduru/StoryDiffusion-jupyter

https://github.com/camenduru/PuLID-jupyter

https://github.com/camenduru/IDM-VTON-jupyter

https://github.com/camenduru/HQEdit-jupyter

https://github.com/camenduru/zest-jupyter

https://github.com/camenduru/Perturbed-Attention-Guidance-jupyter

https://github.com/camenduru/Arc2Face-jupyter

https://github.com/camenduru/champ-jupyter

https://github.com/camenduru/ReNoise-Inversion-jupyter

https://github.com/camenduru/SemanticPalette-jupyter

https://github.com/camenduru/img2img-turbo-jupyter

https://github.com/camenduru/VisualStylePrompting-jupyter

https://github.com/camenduru/TCD-jupyter

https://github.com/camenduru/Multi-LoRA-Composition-jupyter

https://github.com/camenduru/OOTDiffusion-jupyter

https://github.com/camenduru/SDXL-Lightning-jupyter

https://github.com/camenduru/stable-cascade-jupyter

https://github.com/camenduru/ml-mgie-jupyter

https://github.com/camenduru/InstructIR-jupyter

https://github.com/camenduru/InstantID-jupyter

https://github.com/camenduru/PhotoMaker-colab

https://github.com/camenduru/Moore-AnimateAnyone-colab

https://github.com/camenduru/ccsr-colab

https://github.com/camenduru/HandRefiner-colab

https://github.com/camenduru/AnyText-colab

https://github.com/camenduru/normal-depth-diffusion-colab
https://github.com/camenduru/DiffMorpher-colab
https://github.com/camenduru/PASD-colab
https://github.com/camenduru/inferencebot
https://github.com/camenduru/StreamDiffusion-colab
https://github.com/camenduru/UDiffText-colab
https://github.com/camenduru/PatchFusion-colab
https://github.com/camenduru/Mix-of-Show-colab
https://github.com/camenduru/SyncDiffusion-colab
https://github.com/camenduru/DemoFusion-colab
https://github.com/camenduru/playground-colab
https://github.com/camenduru/DeepCache-colab
https://github.com/camenduru/style-aligned-colab
https://github.com/camenduru/MagicAnimate-colab
https://github.com/camenduru/sdxl-turbo-colab
https://github.com/camenduru/cross-image-attention-colab
https://github.com/camenduru/sliders-colab
https://github.com/camenduru/SD-T2I-360PanoImage-colab
https://github.com/camenduru/SSD-1B-colab
https://github.com/camenduru/latent-consistency-model-colab
https://github.com/camenduru/DiffSketcher-colab
https://github.com/camenduru/FreeU-colab
https://github.com/camenduru/stable-fast-colab
https://github.com/camenduru/trainer
https://github.com/camenduru/litelama-colab
https://github.com/camenduru/background-replacement-colab
https://github.com/camenduru/IllusionDiffusion-colab
https://github.com/camenduru/Wuerstchen-colab
https://github.com/camenduru/T2I-Adapter-SDXL-colab
https://github.com/camenduru/facechain-colab
https://github.com/camenduru/StableVideo-colab
https://github.com/camenduru/StableSR-colab
https://github.com/camenduru/Idm3d-colab
https://github.com/camenduru/PixelFusion-colab
https://github.com/camenduru/UniControl-colab
https://github.com/camenduru/tiny-stable-diffusion-colab
https://github.com/camenduru/fabric-colab
https://github.com/camenduru/kohya_ss-colab
https://github.com/camenduru/PSLD-colab
https://github.com/camenduru/control-a-video-colab

https://github.com/camenduru/Matting-Anything-colab

https://github.com/camenduru/TextDiffuser-colab

https://github.com/camenduru/StableStudio-colab

https://github.com/camenduru/Radiata-colab

https://github.com/camenduru/DeepFloyd-IF-colab

https://github.com/camenduru/ControlNet-v1-1-nightly-colab

https://github.com/camenduru/kandinsky-colab

https://github.com/camenduru/stable-diffusion-dreambooth-colab

https://github.com/camenduru/stable-diffusion-diffusers-colab

https://github.com/camenduru/converter-colab

🪅 **ppDiffusers**

https://github.com/camenduru/paddle-ppdiffusers-webui-aistudio-colab

https://github.com/camenduru/paddle-converter-colab

https://github.com/camenduru/paddle-ppdiffusers-webui-aistudio

👀 **Vision LLM**

https://github.com/camenduru/Qwen2-VL-jupyter

https://github.com/camenduru/joy-caption-jupyter

https://github.com/camenduru/LLaVA-OneVision-jupyter

https://github.com/camenduru/MiniGPT4-video-jupyter

https://github.com/camenduru/MoE-LLaVA-jupyter

https://github.com/camenduru/ugen-image-captioning-colab

https://github.com/camenduru/ShareGPT4V-colab

https://github.com/camenduru/MiniGPT-v2-colab

https://github.com/camenduru/LLaVA-colab

https://github.com/camenduru/Qwen-VL-Chat-colab

https://github.com/camenduru/kosmos-2-colab

https://github.com/camenduru/Video-LLaMA-colab

https://github.com/camenduru/MiniGPT-4-colab

🦙 **LLM**

https://github.com/camenduru/nvidia-ngc-jupyter

https://github.com/camenduru/Qwen-Audio-Chat-colab

https://github.com/camenduru/Mistral-colab

https://github.com/camenduru/Llama-2-Onnx-colab

https://github.com/camenduru/japanese-text-generation-webui-colab

https://github.com/camenduru/DoctorGPT-colab

https://github.com/camenduru/Replit-v1-CodeInstruct-3B-colab

https://github.com/camenduru/guanaco-colab

https://github.com/camenduru/nvidia-llm-colab

https://github.com/camenduru/text-generation-webui-colab

https://github.com/camenduru/gpt4all-colab

https://github.com/camenduru/alpaca-lora-colab

Segmentation ML Papers

https://github.com/camenduru/evf-sam2-jupyter

https://github.com/camenduru/sam2-remove-video-background-jupyter

https://github.com/camenduru/YoloWorld-EfficientSAM-jupyter

https://github.com/camenduru/YOLO-World-jupyter

https://github.com/camenduru/EfficientSAM-jupyter

https://github.com/camenduru/OneFormer-colab

https://github.com/camenduru/EdgeSAM-colab

https://github.com/camenduru/SlimSAM-colab

https://github.com/camenduru/FastSAM-colab

https://github.com/camenduru/sam-hq-colab

https://github.com/camenduru/grounded-segment-anything-colab

ML Papers

https://github.com/camenduru/BiRefNet-jupyter

https://github.com/camenduru/EvTexture-jupyter

https://github.com/camenduru/HairFastGAN-jupyter

https://github.com/camenduru/APISR-jupyter

https://github.com/camenduru/bria-rmbg-jupyter

https://github.com/camenduru/autocaption-colab

https://github.com/camenduru/DDColor-colab

https://github.com/camenduru/disco-colab

https://github.com/camenduru/daclip-uir-colab

https://github.com/camenduru/DiffBIR-colab

https://github.com/camenduru/inst-inpaint-colab

https://github.com/camenduru/anime-face-detector-colab

https://github.com/camenduru/insightface-person-detection-colab

https://github.com/camenduru/insightface-face-detection-colab

https://github.com/camenduru/FreeDrag-colab

https://github.com/camenduru/DragDiffusion-colab

https://github.com/camenduru/DragGAN-colab

https://github.com/camenduru/UserControllableLT-colab

https://github.com/camenduru/controlnet-colab

https://github.com/camenduru/notebooks

Models

https://dagshub.com/StyleTTS2/bucilianus-1

https://github.com/camenduru/xenmon-xl-model-colab

https://github.com/camenduru/ios-emoji-xl-model-colab

https://github.com/camenduru/text-to-video-model

Tutorials

https://github.com/camenduru/Text-To-Video-Finetuning-colab

https://github.com/camenduru/train-text-to-image-tpu-tutorial

🧿 **Inference UI**

https://github.com/camenduru/web

https://github.com/camenduru/discord

https://github.com/camenduru/dispatcher

https://github.com/camenduru/scheduler

🎪 **Web UI**

https://github.com/camenduru/stable-diffusion-webui-colab

https://github.com/camenduru/sdxl-colab

https://github.com/camenduru/stable-diffusion-webui-sagemaker

https://github.com/camenduru/stable-diffusion-webui-paperspace

https://github.com/camenduru/stable-diffusion-webui-colab/tree/drive

https://github.com/camenduru/stable-diffusion-webui-colab/tree/training

https://github.com/camenduru/stable-diffusion-webui-docker

https://github.com/camenduru/stable-diffusion-webui-huggingface

https://github.com/camenduru/stable-diffusion-webui-kaggle

https://github.com/camenduru/stable-diffusion-webui-runpod

https://github.com/camenduru/stable-diffusion-webui-vultr

https://github.com/camenduru/stable-diffusion-webui-portable

https://github.com/camenduru/stable-diffusion-webui-artists-to-study

https://github.com/camenduru/stable-diffusion-webui-scripts

https://github.com/camenduru/ai-creator-archive

https://github.com/camenduru/aica

🍥 **Comfy UI to 🍊 Jupyter**

https://github.com/camenduru/sdxl-line-art-style-transfer-jupyter

https://github.com/camenduru/flux-jupyter

https://github.com/camenduru/Animefy

https://github.com/camenduru/SD3-jupyter

https://github.com/camenduru/InstantID-IPAdapter-ControlNet-jupyter

https://github.com/camenduru/IPAdapter-jupyter

https://github.com/camenduru/comfyui-colab

🍇 **RunPod Templates**

https://github.com/camenduru/liveportrait-runpod

https://github.com/camenduru/flux-runpod

🍨 **InvokeAI**

https://github.com/camenduru/InvokeAI-colab

🍭 **Fooocus**

https://github.com/camenduru/Fooocus-docker

https://github.com/camenduru/Fooocus-colab

🧃 **Lambda Labs Demos**

https://github.com/camenduru/fabric-lambda

https://github.com/camenduru/text-generation-webui-lambda

https://github.com/camenduru/llama-2-70b-chat-lambda

https://github.com/camenduru/comfyui-lambda

https://github.com/camenduru/MusicGen-lambda

https://github.com/camenduru/whisper-jax-lambda

https://github.com/camenduru/stable-diffusion-webui-lambda

https://github.com/camenduru/stable-diffusion-webui-api-lambda

https://github.com/camenduru/Radiata-lambda

https://github.com/camenduru/DeepFloyd-IF-lambda

https://github.com/camenduru/falcon-40b-instruct-lambda

https://github.com/camenduru/MiniGPT-4-lambda

https://github.com/camenduru/guanaco-lambda

https://github.com/camenduru/pygmalion-7b-text-generation-webui-lambda

https://github.com/camenduru/one-shot-talking-face-lambda

https://github.com/camenduru/guanaco-13b-lambda

https://github.com/camenduru/guanaco-33b-4bit-lambda

https://github.com/camenduru/converter-lambda

🪶 **Saturn Cloud Demos**

https://github.com/camenduru/stable-diffusion-webui-saturncloud

https://github.com/camenduru/text-generation-webui-saturncloud

https://github.com/camenduru/comfyui-saturncloud

🐚 **Open X Lab Demos**

https://github.com/camenduru/flux-openxlab

https://github.com/camenduru/PhotoMaker-hf/tree/openxlab

https://github.com/camenduru/MotionCtrl-hf/tree/openxlab

https://github.com/camenduru/stable-video-diffusion-openxlab

https://github.com/camenduru/DiffBIR-openxlab

https://github.com/camenduru/I2VGen-XL-openxlab

https://github.com/camenduru/DoctorGPT-openxlab

https://github.com/camenduru/stable-diffusion-webui-openxlab

🦗 **Modal Demos**

https://github.com/camenduru/jupyter-modal

https://github.com/camenduru/DiffBIR-modal

https://github.com/camenduru/stable-diffusion-webui-modal

https://github.com/camenduru/One-2-3-45-modal

🕸️ **Replicate Demos**

https://github.com/camenduru/StoryDiffusion-replicate

https://github.com/camenduru/comfyui-ipadapter-latentupscale-replicate

https://github.com/camenduru/colorize-line-art-replicate

https://github.com/camenduru/HairFastGAN-replicate

https://github.com/camenduru/zest-replicate

https://github.com/camenduru/InstantMesh-replicate

https://github.com/camenduru/MagicTime-replicate

https://github.com/camenduru/mixtral-8×22b-instruct-v0.1-replicate

https://github.com/camenduru/wizardlm-2-8×22b-replicate

https://github.com/camenduru/zephyr-orpo-141b-a35b-v0.1-replicate

https://github.com/camenduru/mixtral-8×22b-v0.1-instruct-oh-replicate

https://github.com/camenduru/mixtral-8×22b-v0.1-4bit-replicate

https://github.com/camenduru/StreamingT2V-replicate

https://github.com/camenduru/EMAGE-replicate

https://github.com/camenduru/MiniGPT4-video-replicate

https://github.com/camenduru/Open-Sora-Plan-replicate

https://github.com/camenduru/attribute-control-replicate

https://github.com/camenduru/Arc2Face-replicate

https://github.com/camenduru/GRM-replicate

https://github.com/camenduru/AniPortrait-vid2vid-replicate

https://github.com/camenduru/GeoWizard-replicate

https://github.com/camenduru/champ-replicate

https://github.com/camenduru/ReNoise-Inversion-replicate

https://github.com/camenduru/open-sora-replicate

https://github.com/camenduru/animatediff-lightning-replicate

https://github.com/camenduru/APISR-replicate

https://github.com/camenduru/DynamiCrafter-interpolation-320×512-replicate

https://github.com/camenduru/VisualStylePrompting-replicate

https://github.com/camenduru/CRM-replicate

https://github.com/camenduru/TripoSR-replicate

https://github.com/camenduru/DSINE-replicate

https://github.com/camenduru/dust3r-replicate

https://github.com/camenduru/MagicDance-replicate

https://github.com/camenduru/stable-cascade-replicate

https://github.com/camenduru/ml-mgie-replicate

https://github.com/camenduru/LGM-ply-to-glb-replicate

https://github.com/camenduru/LGM-replicate

https://github.com/camenduru/metavoice-replicate

https://github.com/camenduru/HandRefiner-replicate

https://github.com/camenduru/bria-rmbg-replicate

https://github.com/camenduru/DynamiCrafter-576×1024-replicate

https://github.com/camenduru/AnimateLCM-replicate

https://github.com/camenduru/MoE-LLaVA-replicate

https://github.com/camenduru/one-shot-talking-face-replicate

https://github.com/camenduru/MotionDirector-replicate

https://github.com/camenduru/DynamiCrafter-replicate

🕹️ **Unreal Engine**

https://github.com/camenduru/unreal-engine-puzzle-collection-blueprint

https://www.unrealengine.com/marketplace/en-US/profile/camenduru

🎮 **Unity**

https://github.com/camenduru/seamless

🏆 **Diffusion Awards**

https://github.com/camenduru/DiffusionAwards

🥽 **VR**

Full Body Motion Capture With HTC Vive and Unity

https://www.youtube.com/watch?v=gE3E-AN7Qiw

👾 **Study**

https://github.com/camenduru/HoudiniStudy

https://github.com/camenduru/BlenderStudy

https://github.com/camenduru/GrasshopperStudy

https://github.com/camenduru/UEStudy

Unity + AI Integrations

Sentis

Top Projects

1. **Unity Sentis Package** - Official Unity Sentis
2. **Sentis Samples** - Reference implementations
3. Hand Pose Sentis - Hand tracking implementation
4. BodyPixSentis

    https://github.com/keijiro/BodyPixSentis

https://colab.research.google.com/drive/1QFGTgdCWmk3BrS7_x6S7xHr7SANp2_Uo - BodyPix Tensorflow to ONNX

Top Models/Model Zoos

1. **Unity Sentis Models** - Official model collection
2. **Ultralytics Models** - Compatible YOLO models
3. **MiDaS Depth Estimation** - Popular depth estimation models

4. **ONNX Model Zoo** - Official collection of pre-trained models
5. **Ultralytics YOLOv8** - State-of-art object detection
6. **Hugging Face ONNX Models** - Diverse ML model collection

Top Training Tools

1. **Unity ML-Agents** - Unity's reinforcement learning framework
2. **PyTorch Image Models** - Training image models
3. **TensorFlow Model Garden** - TensorFlow model collection
4. **Ultralytics YOLOv8 Training** - Latest object detection training
5. **PyTorch ONNX Export Tutorial** - Standard conversion workflow
6. **TensorFlow to ONNX Conversion** - TensorFlow model conversion

as of 3/11/25

AI battle plan & roadmap taking shape. Latest research & recs below. 8pm meet helped especially after recent deep dive - thanks guys. Eager to get others' input & add more of this next sprint -JT

⭐🏆⭐ ONNX Runtime?? ⭐🏆⭐

Balancing speed, integration effort, model availability, long-term stability, and community support, **ONNX Runtime** is currently #1 pick, offering best overall option in short/medium term.

#1 Pick : Best Balance of Speed, Community, Long Term Viability

- **Speed**: Near-native performance with hardware acceleration across platforms
- **Model Availability**: Supports models from virtually all ML frameworks (TensorFlow, PyTorch, scikit-learn, etc.)
- **Community**: Backed by Microsoft, Meta, and other major tech companies with an active open-source community
- **Long-term Stability**: Industry standard with a clear roadmap and widespread adoption
- **Integration Effort**: Moderate - more work than Sentis but manageable with Unity plugins

Implementation Approaches :

1. **Use with Unity Package**: Several community-maintained Unity packages simplify ONNX Runtime integration
2. **Consider Hybrid Approach**: Use Sentis as a fallback or for simpler models where integration speed is critical
3. **Leverage Pre-trained Models**: Huge ecosystem of pre-trained ONNX models available on ONNX Model Zoo and Hugging Face

Comparing Alternatives:

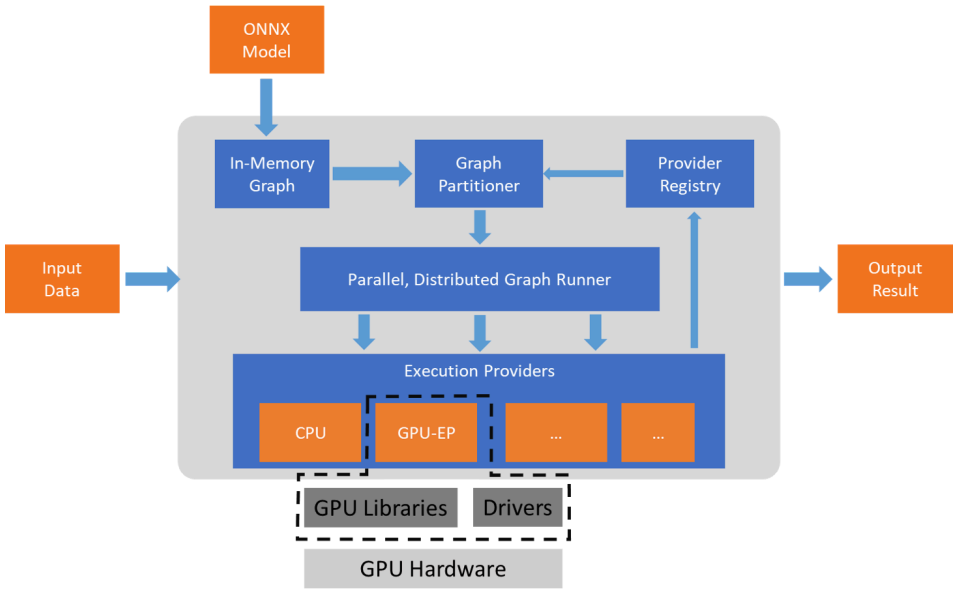
- **Sentis**: Easier integration but fewer models and less community support
- **MediaPipe**: Limited to specific use cases despite easy integration
- **Native Plugins**: Too much integration effort despite excellent performance
- **CoreML**: Platform-limited despite excellent performance on Apple devices

BOTTOM LINE

For Unity XR development with a focus on cross-platform deployment and access to the widest range of AI models, ONNX Runtime provides the most balanced approach with reasonable integration effort and excellent performance characteristics.

NOW ON FENCE DUE TO BELOW??

BUUT ... does very thing that makes it fast - tapping into GPU hardware on phones - via different "Execution Providers" on iOS (CoreML) vs Droid (NNAPI) - mean we need 2 implementations - 1 for each ? Also we haven't found examples yet of how to connect ONNX Runtime + ARfoundation



MediaPipe + AR Foundation examples exist (eg. via Justin) which does implement an a sync strategy, but unclear if this is optimal or will act as a bottleneck as we throttle up Machine Learning and AR features. Also, media pipe integration may break and require additional work for future AR Foundation upgrades...

Open CV + ARFoundation on the other hand have a history of working well together in a small but devoted community of those devoted to keeping it working in unity with each AR foundation upgrade as has been seen over the last 10 to 15 years with the Enoch plug-in ecosystem:

https://github.com/EnoxSoftware/OpenCVForUnity

https://github.com/EnoxSoftware/ARFoundationWithOpenCVForUnityExample

Performance Comparison : Key Metrics

- **ONNX Runtime**: Highest raw performance (30-50% faster than alternatives on iOS via CoreML; 20-40% faster on flagship Droid devices via NNAPI)...BUT requires platform-specific implementations; widest model ecosystem but limited AR Foundation integration examples.
- **Unity Sentis**: Native Unity solution, seamless AR Foundation integration and cross-platform compatibility; performance is consistent but 15-30% slower than ONNX Runtime; offers simplest development experience with the most predictable

behavior across device types.

- **MediaPipe**: Excellent pre-built solutions for vision tasks with existing AR Foundation examples; 10-30% slower than ONNX Runtime on iOS and 5-15% slower on Android flagships, but sometimes faster on mid-range Android devices; may require maintenance with AR Foundation updates.
- **OpenCV for Unity**: Battle-tested solution with 10+ years of updates; performance varies by task (typically 10-25% slower than ONNX Runtime but more consistent across devices); single API across platforms with proven stability and AR Foundation integration history.

🤖🤖 CODE, MODELS, TRAINING 🤖🤖

Top 3 Overview

ONNX Runtime offers the best balance of performance, model ecosystem and cross-platform support. Sentis provides the smoothest Unity integration with excellent cross-platform Unity compatibility. MediaPipe excels at pre-built vision solutions with minimal implementation effort.

1. ONNX Runtime

Top Projects

1. **Microsoft ONNX Runtime** - Official ONNX Runtime repository
2. **Unity Barracuda** - Unity's neural network inference library
3. **YOLOv8 Unity** - Production-ready YOLOv8 implementation

Top Models/Model Zoos

1. **ONNX Model Zoo** - Official collection of pre-trained models
2. **Ultralytics YOLOv8** - State-of-art object detection
3. **Hugging Face ONNX Models** - Diverse ML model collection

Top Training Tools

1. **Ultralytics YOLOv8 Training** - Latest object detection training
2. **PyTorch ONNX Export Tutorial** - Standard conversion workflow
3. **TensorFlow to ONNX Conversion** - TensorFlow model conversion

2. Sentis

Top Projects

1. **Unity Sentis Package** - Official Unity Sentis
2. **Sentis Samples** - Reference implementations
3. Hand Pose Sentis - Hand tracking implementation

Top Models/Model Zoos

1. **Unity Sentis Models** - Official model collection
2. **Ultralytics Models** - Compatible YOLO models
3. **MiDaS Depth Estimation** - Popular depth estimation models

Top Training Tools

1. **Unity ML-Agents** - Unity's reinforcement learning framework
2. **PyTorch Image Models** - Training image models
3. **TensorFlow Model Garden** - TensorFlow model collection

3. MediaPipe

Top Projects

1. **MediaPipe Unity Plugin** - Comprehensive Unity integration
2. **Google MediaPipe** - Official MediaPipe repository
3. **MediaPipe Solutions** - Pre-built solution documentation

Top Models/Model Zoos

1. **MediaPipe Models** - Official models
2. **MediaPipe Tasks Models** - Task-specific models
3. **BlazeFace, BlazePose, and BlazePalm** - Core vision models

Top Training Tools

1. **MediaPipe Model Maker** - Official training workflow
2. **TensorFlow Lite Model Maker** - Compatible model training
3. **Google Colab Tutorials** - Official examples

4. OpenCV

https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088

Cross Platform : multiple platforms, allowing for app development on major platforms.
Upgrades : Enox has upgraded for latest version of OpenCV/Unity for 10+ years
OpenCV Java API : Clone of OpenCV Java, so uses exact same API as OpenCV Java
Easy to Use : Helper functions for interconversion of Unity's Texture2D and OpenCV's Mat. Many classes implement IDisposable, allowing you to manage resources using the "using" statement.
AR VR MR : Works in Augmented Reality, Virtual Reality, and Mixed Reality technologies.
Lots of Example Scenes

Works with Deep Learning models via Dnn module, including various frameworks such as ONNX, TensorFlow, caffe, Torch, Darknet, and more.
Use of WebCamTexture : Allows easier testimg & real-time image processing in Unity Editor
Visual Scripting Support : If we ever turn it back on...

1. ONNX Runtime

Top Projects

1. **Microsoft ONNX Runtime Unity Integration** - Official Unity plugin from Microsoft
2. **Unity Computer Vision ONNX Models** - Unity's computer vision package
3. **Keijiro's ONNX Unity Collection** - Production-ready AR implementations

Top Models/Model Zoos

1. **ONNX Model Zoo** - Official collection of pre-trained models
2. **Ultralytics YOLOv8 ONNX Models** - State-of-art object detection
3. **MobileViT-v2 ONNX** - Fast mobile-optimized vision transformer\

Top Training Tools

1. **YOLOv8 Training Colab** - Latest object detection training
2. **PyTorch to ONNX Conversion** - Standard conversion workflow
3. **Hugging Face Models to ONNX** - Converting latest foundation models

2. Sentis

Top Projects

1. **Official Unity Sentis Samples** - Reference implementations
2. **Sentis AR Foundation Hand Tracking** - Production-ready hand tracking
3. **Unity Sentis Model Gallery** - Optimized model collection

Top Models/Model Zoos

1. **Unity Sentis Model Gallery** - Official model collection
2. **YOLOv9-S ONNX** - Latest YOLO architecture
3. **MobileClip ONNX** - Vision-language models for mobile

Top Training Tools

1. **Unity Sentis Model Training Guide** - Official workflow
2. **YOLOv8 Custom Training Colab** - Custom detection training
3. **Stable Diffusion for Sentis** - Image generation models

3. MediaPipe

Top Projects

1. **MediaPipeUnity Plugin** - Most comprehensive Unity integration
2. **Google MediaPipe Unity Tasks** - Official Unity support
3. **MediaPipe + AR Foundation Integration** - AR-specific implementation

Top Models/Model Zoos

1. **Google MediaPipe Solutions** - Official pre-built solutions
2. **MediaPipe Face Landmarker** - Latest face tracking
3. **MediaPipe Holistic** - Full-body tracking solution

Top Training Tools

1. **MediaPipe Model Maker Colab** - Official training workflow
2. **MediaPipe Custom Gesture Recognizer** - Hand gesture training
3. **MediaPipe Custom Solutions Training** - Advanced customization

More Code : Most Recent, Most Popular

1. ONNX Runtime

Most Recent Projects & Code

1. **Unity ONNX Runtime Execution Provider**
2. **Unity ONNX Barracuda Importer**
3. **AR Foundation + ONNX Runtime Object Detection**
4. **Unity ML Agents with ONNX Runtime**
5. **ONNX Runtime Unity Speech Recognition**
6. **ONNX.Unity - Latest Integration Package**
7. **Unity AR + ONNX Face Detection**
8. **Unity ONNX Runtime Mobile GPU Demo**
9. **AR Foundation with ONNX Runtime Image Classification**
10. **ONNX Runtime Unity WebGL Integration**

Most Widely Supported

1. **Microsoft ONNX Runtime Unity Integration**
2. **Unity Barracuda ONNX Converter**
3. **ONNX.Unity Package**
4. **NNEF-Tools Unity ONNX Support**
5. **Unity AR Foundation ONNX Sample**
6. **Unity ML Toolkit**
7. **XRTK ONNX Runtime Integration**
8. **Unity OpenXR ML Toolkit**
9. **MRTK ONNX Integration**
10. **Ultraleap Hand Tracking ONNX**

Most Popular

1. **Unity Computer Vision ONNX Models**
2. **Unity Perception ONNX Integration**
3. **ONNX Unity Object Detection Sample**
4. **AR Foundation HandTracking with ONNX**
5. **Unity ONNX Runtime Example Asset**
6. **AR Foundation Semantic Segmentation ONNX**
7. **Mixed Reality ONNX Toolkit**
8. **ONNX Runtime Unity Mobile Sample**
9. **Unity AR Foundation YOLOv8 ONNX**
10. **Unity ONNX Vision Sample**

2. Sentis

Most Recent Projects & Code

1. **Official Unity Sentis GitHub**
2. **Unity Sentis AR Foundation Demo**
3. **Sentis Object Recognition in AR**
4. **Unity Sentis ML Model Gallery**
5. **Sentis AR Foundation Face Tracking**
6. **Sentis + AR Foundation Hand Pose Estimation**
7. **Unity Sentis Text Recognition**
8. **Unity Sentis Image Generation**
9. **Unity Sentis AR Character Animation**
10. **Sentis + AR Foundation Augmented Faces**

Most Widely Supported

1. **Unity Sentis Package**
2. **Unity AR Foundation Sentis Integration**
3. **Unity Sentis ONNX Model Importer**
4. **Unity Sentis Model Converter**
5. **Unity Sentis Compute Shaders**

6. **Sentis URP Integration**
7. **Sentis XR Integration Framework**
8. **Sentis Burst Compute**
9. **Unity Sentis Mobile Optimizations**
10. **Unity Sentis Performance Profiler**

Most Popular

1. **Unity Sentis Documentation**
2. **Official Sentis Sample Projects**
3. **Keijiro's Sentis Visual Effects**
4. **AR Foundation Sentis ImageTracker**
5. **Sentis Asset Store Demo**
6. **Sentis Model Zoo**
7. **Sentis AR Foundation Person Segmentation**
8. **Sentis Forum Tutorials**
9. **Sentis MobileNet Sample**
10. **Unity Sentis AR Portal Demo**

3. MediaPipe

Most Recent Projects & Code

1. **Unity MediaPipe Wrapper**
2. **MediaPipe + AR Foundation Integration**
3. **AR Foundation MediaPipe Face Mesh**
4. **MediaPipe Unity Hand Tracking**
5. **MediaPipe Holistic AR Sample**
6. **MediaPipe Unity Pose Tracking**
7. **MediaPipe Unity Object Detection**
8. **Google MediaPipe Tasks for Unity**
9. **MediaPipe Vision + Unity ARKit**
10. **MediaPipe Unity Web Sample**

Most Widely Supported

1. **MediaPipeUnity Plugin**
2. **MediaPipe Tasks Unity Integration**
3. **MediaPipe Holistic Solutions**
4. **MediaPipe AR Foundation Bridge**
5. **Google MediaPipe Assets**
6. **MediaPipe Unity Camera Input**
7. **MediaPipe Unity Sample Models**
8. **MediaPipe TFLite Models**
9. **MediaPipe Object Detection Models**
10. **MediaPipe Face Models**

Most Popular

1. **MediaPipe Unity Plugin**
2. **Google MediaPipe Unity Documentation**
3. **MediaPipe Unity Face Detector**
4. **MediaPipe AR Foundation Face Mesh**
5. **MediaPipe Unity Hand Tracking Asset**
6. **MediaPipe Unity Tutorial Repo**
7. **MediaPipe + Unity AR Demo Collection**
8. **MediaPipe Iris Tracking Demo**
9. **MediaPipe Forum Unity Integration**
10. **MediaPipe Fitness AR App Example**

Architectures & Rankings Explained

1. ONNX Runtime (Speed: 8/10, Cross-Platform: 7/10)

Basic Summary:

ONNX Runtime is a cross-platform inference engine for ONNX models. It loads pre-trained models from any ML framework (converted to ONNX format), optimizes them for the target hardware, and executes them efficiently.

Architecture Flow:

```
CopyInsert[ML Framework (PyTorch, TensorFlow, etc.)] -----> [ONNX Converter] -----> [ONNX Model (.onnx)]
    |
    v
[Unity C# Code] -----> [ONNX Runtime for Unity] -----> [Hardware-specific Execution Provider]
    |                                  |
    v                                  v
[Input Data] -----> [Model Inference] <------ [GPU/CPU/NPU Acceleration]
    |
    v
[Output Predictions] -----> [Unity Gameplay/AR Experience]
```

Key Components:

- Model Converter: Transforms models from various frameworks to ONNX format
- Execution Providers: Hardware-specific backends (CUDA, DirectML, CoreML, etc.)
- Graph Optimization: Automatic optimizations for target hardware
- C# API: Interface between Unity and ONNX Runtime

2. Sentis (Speed: 7/10, Cross-Platform: 9/10)

Basic Summary:

Sentis is Unity's native neural network inference system, designed specifically for Unity projects. It imports ONNX models and executes them using Unity's Compute Shader infrastructure.

Architecture Flow:

```
CopyInsert[ONNX Model (.onnx)] -----> [Sentis Importer] -----> [Unity Asset (.sentis)]
    |
    v
[Unity C# Code] -----> [Sentis Runtime] -----> [Unity Compute Shaders]
    |                                  |
    v                                  v
[Unity Texture/Buffer] -----> [Model Inference] <----- [GPU Acceleration]
    |
    v
[Inference Results] -----> [Unity GameObjects/Components]
```

Key Components:

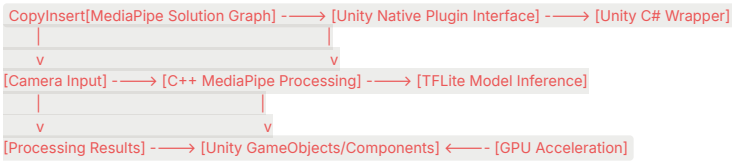
- Sentis Importer: Converts ONNX models to Unity-optimized assets
- Compute Shader Backend: Leverages Unity's existing GPU pipeline
- Tensor API: Unity-native tensor operations
- Unity Editor Integration: Visual debugging tools

3. MediaPipe (Speed: 8/10, Cross-Platform: 6/10)

Basic Summary:

MediaPipe is Google's framework for building ML pipelines, especially for vision tasks. It uses pre-built "Solutions" (hand tracking, face mesh, etc.) implemented as graph-based pipelines.

Architecture Flow:



Key Components:

- Solution Graphs: Pre-configured ML pipelines for specific tasks
- Native Plugin Bridge: C++ to C# communication layer
- TFLite Models: Optimized models running inside MediaPipe
- Camera Input System: Handles device camera integration

Summary

Speed Ranking:

1. **ONNX Runtime (8/10)**: Excellent raw performance with multiple hardware-specific optimizations
2. **MediaPipe (8/10)**: Highly optimized for specific tasks, especially on mobile
3. **Sentis (7/10)**: Good performance within Unity ecosystem but less optimized than specialized solutions

Cross-Platform Migration Ranking:

1. **Sentis (9/10)**: Native Unity solution works identically across all Unity-supported platforms
2. **ONNX Runtime (7/10)**: Good cross-platform support but requires platform-specific configurations
3. **MediaPipe (6/10)**: Requires platform-specific builds and has more limited platform coverage

Key Integration Decisions:

- For **on-device ML in Unity games/apps**: Sentis provides the smoothest developer experience
- For **maximum performance or custom models**: ONNX Runtime offers the best optimization options
- For **pre-built computer vision features**: MediaPipe provides the fastest implementation path

Each technology has its own integration pattern, but they all follow the basic flow of: model import → runtime initialization → inference execution → result processing → Unity scene integration.

🧠🧠 ML Models (XR Optimized) 🧠🧠

1. ONNX Runtime Compatible Models

Verified Models Working with Specific Unity Projects

- **YOLOv8 ONNX** - Works with **Unity AR Foundation YOLOv8 ONNX**
- **MobileNet ONNX** - Works with **Unity ONNX Vision Sample**
- **MediaPipe BlazeFace ONNX** - Works with **AR Foundation + ONNX Runtime Object Detection**
- **PoseNet ONNX** - Works with **ONNX Unity Object Detection Sample**

State-of-the-Art Models Optimized for Mobile/XR

1. **MobileViT-v2 ONNX**
2. **EfficientNet-Lite ONNX**
3. **YOLOv8s-Seg ONNX**

4. **MobileBERT ONNX**
5. **BlazePose-ONNX**
6. **MobileNetV3-SSD ONNX**
7. **DeepLabV3-MobileNet ONNX**
8. **Fast Neural Style Transfer ONNX**
9. **MobileSAM ONNX**
10. **WaveRNN ONNX**

Most Recent Mobile-Optimized ONNX Models

1. **YOLOv9-C ONNX**
2. **MobileOne ONNX**
3. **PP-LiteSeg ONNX**
4. **SwinTransformer-Tiny ONNX**
5. **EfficientFormer ONNX**
6. **MobileVLM ONNX**
7. **HRNet-Lite ONNX**
8. **EdgeSpeechNets ONNX**
9. **TinyML ONNX Collection**
10. **FastestDet ONNX**

ONNX Model Zoos & Repositories

1. **ONNX Model Zoo**
2. **Hugging Face ONNX Models**
3. **PINTO ONNX Model Zoo** ***
4. **Ultralytics YOLOv8 ONNX Models**
5. **Microsoft ONNX Models**
6. **OpenVINO ONNX Models**
7. **MediaPipe ONNX Models**
8. **Intel OpenVINO Model Zoo**
9. **TensorFlow Lite to ONNX Models**
10. **ONNX-TensorRT Models**

2. Sentis Compatible Models

Verified Models Working with Specific Unity Projects

- **MobileNet v2 ONNX** - Works with **Sentis MobileNet Sample**
- **YOLOv8n ONNX** - Works with **Sentis Object Recognition in AR**
- **HandPose ONNX** - Works with **Sentis + AR Foundation Hand Pose Estimation**
- **FaceMesh ONNX** - Works with **Sentis + AR Foundation Augmented Faces**

State-of-the-Art Models Optimized for Mobile/XR

1. **EfficientDet-Lite ONNX**
2. **MobileViT ONNX**
3. **YOLOv8n-Seg ONNX**
4. **BiSeNet-Lite ONNX**
5. **MoveNet ONNX**
6. **DistilBERT ONNX**
7. **FastSAM ONNX**
8. **TinyNeRF ONNX**
9. **MobileStyleGAN ONNX**
10. **Mobile Depth Estimation ONNX**

Most Recent Mobile-Optimized Sentis-Compatible Models

1. **YOLOv9-S ONNX**
2. **EfficientViT ONNX**
3. **MobileClip ONNX**
4. **Tiny-ViLT ONNX**
5. **EdgeFormer ONNX**
6. **MobileNeRF ONNX**
7. **TinyML Anomaly Detection ONNX**
8. **LCM-LoRA ONNX**
9. **MobileSR ONNX**
10. **FastSAM-XS ONNX**

Sentis Model Zoos & Repositories

1. **Unity Sentis Model Gallery**
2. **Unity Sentis Sample Models**
3. **Sentis Model Zoo**
4. **Keijiro's Sentis Models**
5. **Unity AR Foundation ML Zoo**
6. **Barracuda to Sentis Model Conversion**
7. **Sentis Mobile Model Zoo**
8. **Sentis Compute Shader Models**
9. **Unity XR ML Models**
10. **Unity Platforms Sentis Models**

3. MediaPipe Compatible Models

Verified Models Working with Specific Unity Projects

- **MediaPipe Face Detection** - Works with **MediaPipe Unity Face Detector**
- **MediaPipe Hand Landmark Model** - Works with **MediaPipe Unity Hand Tracking**
- **MediaPipe Pose Landmark** - Works with **MediaPipe Unity Pose Tracking**
- **MediaPipe Holistic** - Works with **MediaPipe Holistic AR Sample**

State-of-the-Art Models Optimized for Mobile/XR

1. **BlazeFace**
2. **BlazePose**
3. **BlazePalm**
4. **FaceMesh**
5. **MediaPipe Hair Segmentation**
6. **MediaPipe Object Detection**
7. **MediaPipe Iris**
8. **MediaPipe Selfie Segmentation**
9. **MediaPipe Hand Gesture Recognition**
10. **MediaPipe 3D Objectron**

Most Recent Mobile-Optimized MediaPipe Models

1. **MediaPipe Face Landmarker**
2. **MediaPipe Gesture Recognizer**
3. **MediaPipe Pose Landmarker**
4. **MediaPipe Interactive Segmenter**
5. **MediaPipe Text Classification**
6. **MediaPipe Image Embedder**
7. **MediaPipe Hand Landmarker**
8. **MediaPipe Object Detector**

9. **MediaPipe Text Embedder**
10. **MediaPipe Audio Classification**

MediaPipe Model Zoos & Repositories

1. **Google MediaPipe Models**
2. **MediaPipe Solutions Models**
3. **MediaPipe Tasks Models**
4. **MediaPipe Face Models**
5. **MediaPipe Hand Models**
6. **MediaPipe Pose Models**
7. **MediaPipe Holistic Models**
8. **MediaPipe Object Detection Models**
9. **MediaPipe Selfie Segmentation Models**
10. **MediaPipe Tasks Core Models**

Cross-Platform Compatibility Notes

- **ONNX Models**: Generally work across all three frameworks with proper conversion/loading
- **TensorFlow Lite Models**: Native to MediaPipe but can be converted to ONNX for Sentis/ONNX Runtime
- **Model Size Guidelines for XR**:
  - Small: <5MB (ideal for mobile AR)
  - Medium: 5-25MB (suitable for most mobile XR applications)
  - Large: 25-50MB (challenging but possible with optimization)
  - Very Large: >50MB (generally not suitable for mobile XR)

BOTTOM LINE

For best cross-platform compatibility, focus on:

MobileNet, EfficientNet, YOLOv8-nano, and BlazeFace/BlazePose families, which have been extensively optimized for mobile & XR platforms.

🏋️Training & Finetuning Tools (XR-optimized)🏋️

1. ONNX Runtime Compatible Training Tools

Project-Verified Training Resources

- **YOLOv8 Training Colab** - Creates models compatible with **Unity AR Foundation YOLOv8 ONNX**
- **MobileNet Fine-tuning Colab** - Creates models compatible with **Unity ONNX Vision Sample**
- **PoseNet Training Colab** - Creates models compatible with **ONNX Unity Object Detection Sample**

ONNX Training Resources

1. **PyTorch to ONNX Conversion Colab**
2. **TensorFlow to ONNX Conversion Tutorial**
3. **ONNX Model Training & Optimization**
4. **Efficient Object Detection Training**
5. **Hugging Face Models to ONNX Conversion**

6. **ONNX Quantization Tools**
7. **Custom Model Training for ONNX Export**
8. **ONNX Optimization with ONNX Runtime**
9. **Transfer Learning with ONNX Export**
10. **Mobile-Optimized ONNX Models Training**

2. Sentis Compatible Training Tools

Project-Verified Training Resources

- **MobileNetV2 Classification Training** - Creates models compatible with **Sentis MobileNet Sample**
- **YOLOv8 Custom Training Colab** - Creates models compatible with **Sentis Object Recognition in AR**
- **Hand Pose Estimation Training** - Creates models compatible with **Sentis + AR Foundation Hand Pose Estimation**

Sentis Training Resources

1. **Unity Sentis Model Training Guide**
2. **PyTorch to Sentis Conversion Tutorial**
3. **Sentis-Compatible Image Segmentation Training**
4. **Stable Diffusion Model Training for Sentis**
5. **Transfer Learning for Sentis-Compatible Models**
6. **Mobile-Optimized Model Training**
7. **Keijiro's Unity ML Training Tools**
8. **TensorFlow Lite to Sentis Workflow**
9. **Custom Segmentation Models for Sentis**
10. **Image Classification for Unity**

3. MediaPipe Compatible Training Tools

Project-Verified Training Resources

- **MediaPipe Face Detection Training** - Creates models compatible with **MediaPipe Unity Face Detector**
- **MediaPipe Hand Landmark Training** - Creates models compatible with **MediaPipe Unity Hand Tracking**
- **MediaPipe Pose Landmark Training** - Creates models compatible with **MediaPipe Unity Pose Tracking**

MediaPipe Training Resources

1. **MediaPipe Model Maker Colab**
2. **MediaPipe Custom Gesture Recognizer**
3. **MediaPipe Tasks Custom Training**
4. **Face Landmark Model Training**
5. **MediaPipe Object Detection Training**
6. **Custom Pose Estimation Model**
7. **MediaPipe Vision Tasks Training**
8. **Audio Classification with MediaPipe**
9. **BlazeFace Custom Training**
10. **MediaPipe Custom Solutions Training**

Additional Cross-Framework Training Tools

Model Conversion & Optimization

1. **ONNX-TensorFlow Converter**
2. **TensorFlow Lite Converter**
3. **Unity Model Optimization Tool**
4. **Mobile Model Quantization Tools**
5. **ONNX Model Optimization**

Cross-Platform Training Guidelines

- For **maximum compatibility** across all three frameworks:
  1. Train in PyTorch or TensorFlow
  2. Export to ONNX format
  3. Test with mobile-optimized architectures (MobileNet, EfficientNet, YOLO-nano)
  4. Keep model size under 15MB for best mobile/XR performance
  5. Use quantization (int8/fp16) when possible
  6. Test on target hardware before full deployment
- For **Unity-specific optimization**:
  1. Consider input/output tensor shapes that match Unity textures
  2. Test with different precision settings in Sentis/Barracuda
  3. Profile memory usage in Unity while running inference
  4. Consider batching strategies appropriate for real-time applications

These training resources offer workflows to create models that work efficiently within Unity's XR environment, with special attention to mobile and XR hardware constraints.

ML + Unity : Matrix Comparisons

Performance & Integration

Technology	Performance	Unity Integration	Development Complexity	Memory Footprint	Build Size Impact
Sentis	★★★★☆	★★★★★	★★★★★ (Simple)	★★★★☆ (Low)	★★★★☆ (Small)
ONNX Runtime	★★★★★	★★★☆☆	★★★☆☆ (Moderate)	★★★☆☆ (Medium)	★★☆☆☆ (Large)
MediaPipe	★★★★☆	★★★☆☆	★★★☆☆ (Moderate)	★★☆☆☆ (Large)	★★☆☆☆ (Large)
Native Plugins	★★★★★	★★☆☆☆	★☆☆☆☆ (Complex)	★★★☆☆ (Varies)	★★☆☆☆ (Large)
CoreML	★★★★★ (iOS)	★★★☆☆	★★☆☆☆ (Complex)	★★★★☆ (Low)	★★★☆☆ (Medium)

Platform Support & Features

Technology	Cross-Platform	iOS	Android	Web	Desktop	XR Headsets	Pre-built ML Features
Sentis	★★★★★	★★★★☆	★★★★☆	★★★★☆	★★★★★	★★★★★	★★☆☆☆ (Limited)
ONNX Runtime	★★★★☆	★★★★☆	★★★★☆	★★☆☆☆	★★★★★	★★★★☆	★★☆☆☆ (Limited)
MediaPipe	★★★★☆	★★★★☆	★★★★★	★★★★☆	★★★★☆	★★★☆☆	★★★★★ (Extensive)
Native Plugins	★★★☆☆	★★★★★	★★★★★	★☆☆☆☆	★★★★★	★★★★☆	★★★★★ (Custom)
CoreML	★☆☆☆☆	★★★★★	✖	✖	★★★☆☆ (macOS)	★★★☆☆ (Apple)	★★★☆☆ (Good)

Use Cases & Future-Proofing

Technology	AR/VR Apps	Computer Vision	NLP/Audio	Custom Models	Long-term Viability	Update Frequency
Sentis	★★★★★	★★★☆☆	★★★☆☆	★★★★☆	★★★★★	★★★☆☆ (Unity Cycle)
ONNX Runtime	★★★★☆	★★★★☆	★★★★★	★★★★★	★★★★★	★★★★★ (Frequent)
MediaPipe	★★★★☆	★★★★★	★★★☆☆	★★☆☆☆	★★★★☆	★★★★☆ (Active)
Native Plugins	★★★★★	★★★★★	★★★★★	★★★★★	★★★☆☆	★★★★★ (Varies)
CoreML	★★★★☆ (iOS)	★★★★☆	★★★★☆	★★★★☆	★★★☆☆ (Apple Only)	★★★☆☆ (Annual)

Development & Maintenance Considerations

Technology	Documentation	Community Support	Learning Curve	Integration Effort	Maintenance Burden	Unity Version Compatibility
Sentis	★★★★☆	★★★☆☆	★★★★★ (Easy)	★★★★★ (Minimal)	★★★★★ (Low)	★★★★★
ONNX Runtime	★★★★★	★★★★★	★★★☆☆ (Moderate)	★★★☆☆ (Moderate)	★★★☆☆ (Moderate)	★★★☆☆
MediaPipe	★★★★☆	★★★★☆	★★★☆☆ (Moderate)	★★★☆☆ (Moderate)	★★☆☆☆ (High)	★★★☆☆
Native Plugins	★★★☆☆	★★★★☆	★☆☆☆☆ (Difficult)	★☆☆☆☆ (High)	★☆☆☆☆ (Very High)	★★☆☆☆
CoreML	★★★★☆	★★★☆☆	★★★☆☆ (Moderate)	★★☆☆☆ (High)	★★☆☆☆ (High)	★★★☆☆

ML + Unity : List Comparisons

Same as above but simplified as lists

Sentis

- **Strengths:**
  - Native Unity integration
  - Easy to use with minimal code
  - Works consistently across all Unity platforms
  - Small memory and build footprint
  - Best for AR/VR/XR in Unity ecosystem
- **Limitations:**
  - Fewer pre-built ML features
  - Updates tied to Unity release cycle
  - Less flexible for custom ML pipelines

ONNX Runtime

- **Strengths:**
  - Highest raw performance
  - Excellent for custom models

    - Strong community and ecosystem
    - Frequent updates with latest ML features
    - Best for complex ML workloads
- **Limitations:**
    - Requires more integration work
    - Larger build size
    - Less optimized for Unity workflow

MediaPipe

- **Strengths:**
    - Extensive pre-built ML solutions
    - Excellent for computer vision tasks
    - Great for rapid prototyping
    - Strong on mobile platforms
    - Best for face/hand/pose tracking
- **Limitations:**
    - High memory usage
    - More complex integration
    - Higher maintenance burden

Native Plugins (OpenCV, TensorFlow, etc.)

- **Strengths:**
    - Maximum performance potential
    - Full customization capability
    - Platform-specific optimizations
    - Access to cutting-edge ML features
    - Best for specialized use cases
- **Limitations:**
    - Very complex integration
    - Highest maintenance burden
    - Potential compatibility issues
    - Requires platform-specific code

CoreML

- **Strengths:**
    - Best performance on Apple devices
    - Direct access to Apple Neural Engine
    - Privacy-focused on-device processing
    - Excellent iOS integration
    - Best for Apple-exclusive apps

- **Limitations:**
  - Apple platforms only
  - Complex integration with Unity
  - Platform-specific code required
  - Higher maintenance burden

Quick Decision Guide

- **Choose Sentis when:**
  - You want minimal integration effort
  - You need cross-platform consistency
  - You're building primarily in Unity
  - Long-term maintenance is a concern
- **Choose ONNX Runtime when:**
  - Performance is critical
  - You need latest ML operators
  - You have existing ML pipelines
  - You require maximum model flexibility
- **Choose MediaPipe when:**
  - You need pre-built vision features
  - Quick implementation is a priority
  - Mobile performance is important
  - Time-to-market matters most
- **Choose Native Plugins when:**
  - Maximum performance is essential
  - You need very specific optimizations
  - You have resources for maintenance
  - Platform-specific code is acceptable
- **Choose CoreML when:**
  - You're targeting only Apple devices
  - Maximum iOS performance is required
  - On-device processing is critical
  - Privacy is a major concern

Comparing OpenCV, MediaPipe, ONNX Runtime

OpenCV

- **Core Purpose**: A comprehensive computer vision library focused on image processing, video analysis, and classical computer vision algorithms
- **Key Strengths**:
  - Extensive set of traditional computer vision algorithms (feature detection, object tracking, camera calibration)
  - Mature C++ codebase with bindings for multiple languages including C#
  - High-performance image processing operations
  - Hardware acceleration support
- **Unity Integration**: Requires wrappers like OpenCVForUnity or native plugins

MediaPipe

- **Core Purpose**: Google's framework for building multimodal (video, audio, time-series) applied ML pipelines
- **Key Strengths**:
  - Pre-built solutions for hand tracking, face detection, pose estimation, and object detection
  - Graph-based architecture for ML pipeline construction
  - Optimized for mobile devices and real-time applications
  - Cross-platform with mobile-first approach
- **Unity Integration**: Available through Unity packages like MediaPipe Unity Plugin or custom native plugins

ONNX Runtime

- **Core Purpose**: High-performance inference engine for ONNX (Open Neural Network Exchange) models
- **Key Strengths**:
  - Model portability across frameworks (TensorFlow, PyTorch, etc.)
  - Optimized inference across CPU, GPU, and specialized hardware
  - Consistent performance across multiple platforms
  - Focused on efficient neural network execution
- **Unity Integration**: Works with Unity through Barracuda, Sentis, or custom native plugins

Key Differences

1. **Scope**:
   - OpenCV: Broad computer vision library with minimal ML focus
   - MediaPipe: End-to-end ML pipelines with pre-built solutions
   - ONNX Runtime: Purely an inference engine for neural network models
2. **Implementation Approach**:
   - OpenCV: Manual algorithm implementation and customization
   - MediaPipe: Graph-based pipelines with pre-trained models
   - ONNX Runtime: Model-centric approach with framework portability
3. **Mobile/XR Development**:
   - OpenCV: Requires careful optimization for mobile/XR
   - MediaPipe: Built specifically for mobile performance
   - ONNX Runtime: Optimized for cross-platform deployment
4. **Integration Complexity in Unity**:
   - OpenCV: Typically higher integration effort
   - MediaPipe: Moderate integration with built-in solutions
   - ONNX Runtime: Streamlined through Sentis for Unity's neural network inference

For Unity XR applications, these technologies are often used in combination - MediaPipe for real-time tracking, OpenCV for specialized vision processing, and ONNX Runtime for running custom ML models.

CoreML Comparison

CoreML

- **Core Purpose**: Apple's machine learning framework designed specifically for iOS and macOS devices
- **Key Strengths**:
  - Deep integration with Apple hardware (Neural Engine, GPU, CPU)
  - Highly optimized for Apple devices with low power consumption
  - Support for on-device machine learning to preserve privacy
  - Compatible with models from popular frameworks (TensorFlow, PyTorch, scikit-learn) through conversion
  - Built-in support for vision, natural language, and sound analysis
- **Unity Integration**: Available through native plugins or Unity's AR Foundation

CoreML vs Others

CoreML vs OpenCV

- **Platform Focus**: CoreML is Apple-exclusive while OpenCV is cross-platform
- **Optimization**: CoreML offers superior performance on Apple devices through hardware acceleration
- **Scope**: CoreML focuses on ML model execution while OpenCV covers broader computer vision algorithms
- **Integration**: CoreML has native iOS integration; OpenCV requires more custom implementation

CoreML vs MediaPipe

- **Ecosystem**: CoreML is tied to Apple's ecosystem; MediaPipe works across platforms
- **Pre-built Solutions**: Both offer ready-made solutions, but MediaPipe's are platform-agnostic
- **Mobile Optimization**: Both are highly optimized for mobile, but CoreML has an edge on Apple hardware
- **Development Model**: CoreML is model-centric; MediaPipe is pipeline-centric

CoreML vs ONNX Runtime

- **Model Format**: CoreML uses its proprietary format (.mlmodel); ONNX Runtime uses open ONNX format
- **Platform Coverage**: CoreML is Apple-only; ONNX Runtime is cross-platform
- **Hardware Acceleration**: Both leverage hardware effectively, but CoreML is more tightly integrated with Apple silicon
- **Integration Complexity**: CoreML has simpler integration for iOS-only apps; ONNX Runtime offers greater flexibility

CoreML in Unity/XR Development

- Ideal for iOS/macOS AR applications using ARKit
- Offers excellent performance for on-device ML in Apple VR/AR headsets
- Requires platform-specific code paths when targeting multiple platforms
- Works well with AR Foundation in Unity for iOS deployments
- Limited to Apple platforms, requiring alternatives for Android or other XR targets

BOTTOM LINE

When developing cross-platform XR applications in Unity, developers often use CoreML for Apple devices while implementing alternatives (like MediaPipe or ONNX Runtime) for other platforms, requiring careful architecture to manage these platform-specific implementations.

RESULTS: Optimal Scalable Multimodal ML+XR

Core Components & Technologies

1. Model Format & Inference

- **ONNX + ONNX Runtime**:
  - Universal model format supporting nearly all frameworks
  - Strong industry backing (Microsoft, Facebook, AWS, etc.)
  - Excellent cross-platform support and optimization
  - Growing community and tools ecosystem

2. Training Framework

- **PyTorch**:
  - Research-friendly yet production-capable
  - Dynamic computation graph for flexible model architecture
  - Strong community support and industry adoption
  - Seamless ONNX export capabilities

3. Frontend Framework (Unity-based)

- **Unity + Sentis/Barracuda**:
  - For ARVR/XR applications with neural network inference
  - AR Foundation for cross-platform AR capabilities
  - Custom C# wrapper system for platform-specific optimizations

4. Mobile/Edge Optimization

- **TensorRT** (NVIDIA platforms)
- **ONNX Runtime Mobile** (cross-platform)
- **Platform-specific accelerators**:
  - CoreML bridges for iOS
  - Android Neural Networks API bridges
  - Hardware-specific optimizers as abstracted services

5. Platform Connectors

- **REST APIs + gRPC**:
  - For cloud-to-edge communication
  - Protocol Buffers for efficient serialization
  - WebRTC for real-time requirements

6. Data Pipeline & Format

- **Apache Arrow**:
  - Efficient columnar memory format
  - Language-agnostic data interchange
  - Optimized for analytics workloads

7. Deployment & Scalability

- **Kubernetes + Helm**:
  - Container orchestration for cloud deployments
  - Auto-scaling capabilities
  - Reproducible deployments

Architecture Design Principles

1. **Modular Abstraction Layers**:
   - Hardware abstraction layer
   - Model serving abstraction layer
   - Data processing abstraction layer
2. **Hybrid Edge-Cloud Processing**:
   - Edge for latency-sensitive operations
   - Cloud for compute-intensive training and large models
   - Smart workload distribution
3. **Standardized Interfaces**:
   - OpenXR for XR devices
   - WebAssembly for web platform
   - OpenAPI/Swagger for service APIs
4. **Resilient Design Patterns**:
   - Circuit breakers for service degradation
   - Feature toggles for gradual rollout
   - Fallback mechanisms for offline operation

Timeline Considerations

1-Year Horizon

- Focus on ONNX-based pipeline establishment
- Platform-specific optimizations for current hardware
- Basic model training and deployment workflows

3-Year Horizon

- Enhanced on-device ML capabilities
- Automated ML pipelines for continuous improvement
- Advanced multi-modal fusion techniques

5-Year Horizon

- Distributed training across edge devices
- Federated learning incorporation
- Self-optimizing infrastructure

10-Year Horizon

- Platform-agnostic neural interfaces

- Quantum ML integration where beneficial
- Neuromorphic computing adaptation

Implementation Strategy

1. **Start with Open Standards**:
   - ONNX for model interchange
   - OpenXR for XR device interaction
   - Open Neural Network Exchange (ONNX) format
2. **Build Abstraction Layers**:
   - Develop a unified API for all ML operations
   - Create adapters for platform-specific optimizations
   - Implement feature detection for capability-based degradation
3. **Progressive Enhancement**:
   - Core functionality through standard interfaces
   - Advanced features through platform-specific extensions
   - Graceful degradation on less capable devices

This architecture maximizes long-term flexibility while providing immediate practical implementation paths, especially for Unity-based XR applications with cross-platform requirements.

Performance Comparisons

Unity + Sentis

- **Performance**: Generally fastest for on-device inference within Unity
- **Key Advantages**:
  - Native Unity integration with C# API
  - GPU acceleration through Compute Shaders
  - Optimized for Unity's rendering pipeline
  - No native plugin dependencies
- **Best For**: Cross-platform projects requiring consistent performance across devices

Unity + MediaPipe

- **Performance**: Good for specific vision tasks, but introduces overhead
- **Key Advantages**:
  - Pre-built solutions for common tasks (hand tracking, face mesh, etc.)
  - Mobile-optimized algorithms
  - Cross-platform support
- **Disadvantages**:
  - Additional integration complexity
  - Potential memory overhead from separate runtime

- **Best For**: Projects requiring specific pre-built ML features like hand/face tracking

Unity + Native Plugin (OpenCV, TensorFlow, etc.)

- **Performance**: Can be fastest with proper optimization, but varies widely
- **Key Advantages**:
  - Maximum flexibility for specific hardware optimization
  - Access to platform-specific acceleration (SIMD, NEON, etc.)
  - Full control over memory management
- **Disadvantages**:
  - Complex integration and maintenance
  - Platform-specific code paths required
  - Potential for crashes/memory leaks
- **Best For**: Performance-critical applications requiring specific optimizations

Unity + CoreML

- **Performance**: Fastest option for Apple devices
- **Key Advantages**:
  - Direct access to Apple Neural Engine
  - Highly optimized for iOS/macOS
  - Battery-efficient processing
- **Disadvantages**:
  - Apple-only solution
  - Requires platform-specific code
  - Additional integration steps
- **Best For**: iOS/iPadOS/macOS-exclusive AR applications

Unity + AR Foundation with ML Models

- **Performance**: Varies based on implementation (Sentis, native, etc.)
- **Key Advantages**:
  - Unified AR framework across platforms
  - Direct integration with AR subsystems
  - Consistent developer experience
- **Disadvantages**:
  - May not access all platform-specific optimizations
  - Potential overhead from abstraction layer
- **Best For**: Cross-platform AR applications with ML features

Performance Comparison in Different Contexts

Mobile AR Applications (ARKit/ARCore)

1. **Unity + AR Foundation + Sentis**: Best balance of performance and cross-platform
2. **Unity + AR Foundation + CoreML** (iOS) / **Native ML** (Android): Best platform-specific performance
3. **Unity + AR Foundation + MediaPipe**: Best for rapid development with pre-built features

Web-Based 3D/XR (WebXR)

1. **Unity WebGL + WebAssembly + TensorFlow.js**: Most compatible approach
2. **Unity WebGL + Custom JS Bridge + MediaPipe**: Best for vision tasks
3. **Unity WebGL + WebGPU + Custom ML**: Best future-oriented approach

Standalone XR Applications (VR/MR Headsets)

1. **Unity + Native ML Plugin**: Best for performance-critical applications
2. **Unity + Sentis**: Best for development efficiency and maintainability
3. **Unity + MediaPipe**: Best for rapid implementation of common ML features

Key Decision Factors

1. **Device Target Diversity**:
   - Single platform → Native/CoreML
   - Multiple platforms → Sentis or MediaPipe
2. **ML Task Complexity**:
   - Standard vision tasks → MediaPipe
   - Custom ML models → Sentis or Native plugins
3. **Development Resources**:
   - Limited resources → Sentis or MediaPipe
   - Performance-critical → Native implementation
4. **Maintenance Horizon**:
   - Short-term → Platform-specific optimizations
   - Long-term → Unified abstractions (Sentis)

BOTTOM LINE

Sentis offers the best balance of performance and cross-platform consistency within the Unity ecosystem, while platform-specific approaches like CoreML provide maximum performance for single-platform targets. The ideal approach often involves a hybrid strategy, using Sentis as the core with platform-specific optimizations where critical.

Me

Why would you choose sentence over Onyx runtime

Thought Process

The user is asking about the tradeoffs between choosing Sentis (Unity's ML inference solution) versus ONNX Runtime. Let me analyze the key reasons one would choose Sentis over ONNX Runtime in a Unity context.

Why Sentis Over ONNX Runtime in Unity?

Native Unity Integration Advantages

- **Direct Unity Engine Integration**: Sentis is built specifically for Unity by Unity Technologies
- **Compute Shader Utilization**: Leverages Unity's existing GPU compute infrastructure

- **Rendering Pipeline Compatibility**: Designed to work with URP/HDRP without context switching
- **ECS/DOTS Compatibility**: Better alignment with Unity's data-oriented design approach

Development Workflow Benefits

- **C# First Development**: Pure C# API without external dependencies or interop layers
- **Asset Pipeline Integration**: Models treated as Unity assets with inspector preview
- **Project Build System**: Automatic platform-specific optimizations during build
- **Editor Tools**: Visual debugging and profiling tools in Unity Editor

Unity-Specific Optimizations

- **Unity Memory Management**: Uses Unity's memory management system rather than external allocations
- **Burst Compiler Compatibility**: Potential for Burst-compiled operations
- **Render Texture Direct Access**: Efficient processing of camera/render texture data without copies
- **Unity Job System**: Better integration with Unity's multithreading architecture

Platform Considerations

- **WebGL Compatibility**: Better supported in Unity's WebGL export
- **Mobile Performance**: Optimized specifically for Unity's mobile rendering pipeline
- **XR Subsystem Integration**: Direct access to AR/VR camera and sensor data
- **Consistent Cross-Platform Behavior**: Same code works across all Unity-supported platforms

Practical Considerations

- **Simplified Dependency Management**: No need to maintain external native libraries
- **Reduced Binary Size**: Potentially smaller build sizes for mobile/WebGL
- **Unity Version Alignment**: Guaranteed compatibility with Unity version upgrades
- **Unity Support Channels**: Access to Unity-specific support resources

ONNX Runtime Advantages (for comparison)

- Wider ecosystem and tool support
- Potentially more optimized for non-Unity platforms
- More frequent updates for new ML operators
- Better integration with existing ML pipelines

Sentis is generally preferred when deep Unity integration and workflow are prioritized over having the absolute latest ML features or when targeting Unity-centric deployment scenarios, especially in AR/XR contexts where integration with Unity's rendering and physics systems is critical.

ML + Unity: Simplified Comparison

Sentis

- **Strengths:**

- Native Unity integration
- Easy to use with minimal code
- Works consistently across all Unity platforms
- Small memory and build footprint
- Best for AR/VR/XR in Unity ecosystem

- **Limitations:**
  - Fewer pre-built ML features
  - Updates tied to Unity release cycle
  - Less flexible for custom ML pipelines

ONNX Runtime

- **Strengths:**
  - Highest raw performance
  - Excellent for custom models
  - Strong community and ecosystem
  - Frequent updates with latest ML features
  - Best for complex ML workloads
- **Limitations:**
  - Requires more integration work
  - Larger build size
  - Less optimized for Unity workflow

MediaPipe

- **Strengths:**
  - Extensive pre-built ML solutions
  - Excellent for computer vision tasks
  - Great for rapid prototyping
  - Strong on mobile platforms
  - Best for face/hand/pose tracking
- **Limitations:**
  - High memory usage
  - More complex integration
  - Higher maintenance burden

Native Plugins (OpenCV, TensorFlow, etc.)

- **Strengths:**
  - Maximum performance potential
  - Full customization capability
  - Platform-specific optimizations
  - Access to cutting-edge ML features

    - Best for specialized use cases
- **Limitations:**
    - Very complex integration
    - Highest maintenance burden
    - Potential compatibility issues
    - Requires platform-specific code

CoreML

- **Strengths:**
    - Best performance on Apple devices
    - Direct access to Apple Neural Engine
    - Privacy-focused on-device processing
    - Excellent iOS integration
    - Best for Apple-exclusive apps
- **Limitations:**
    - Apple platforms only
    - Complex integration with Unity
    - Platform-specific code required
    - Higher maintenance burden

Quick Decision Guide

- **Choose Sentis when:**
    - You want minimal integration effort
    - You need cross-platform consistency
    - You're building primarily in Unity
    - Long-term maintenance is a concern
- **Choose ONNX Runtime when:**
    - Performance is critical
    - You need latest ML operators
    - You have existing ML pipelines
    - You require maximum model flexibility
- **Choose MediaPipe when:**
    - You need pre-built vision features
    - Quick implementation is a priority
    - Mobile performance is important
    - Time-to-market matters most
- **Choose Native Plugins when:**
    - Maximum performance is essential
    - You need very specific optimizations

- You have resources for maintenance
- Platform-specific code is acceptable

- **Choose CoreML when:**
  - You're targeting only Apple devices
  - Maximum iOS performance is required
  - On-device processing is critical
  - Privacy is a major concern

⭐⭐ONNX Runtime⭐⭐

For balancing speed, integration effort, model availability, long-term stability, and community support, **ONNX Runtime** offers the best overall compromise... see top of page...

Research above is how we compared options & settled on this is #1 choice for further testing.

Ranking as of meeting 3/10/25

Fastest Computer Vision & Deep Learning Options for Unity Cross-Platform Development

Analysis of the fastest options for running computer vision and deep learning models across all our current target platforms (according to research & Claude Sonnet 3.7 as of 3/10/25)

OpenCV vs MediaPipe vs ONNX Runtime

ONNX Runtime, MediaPipe & OpenCV have emerged as top 3 thus far. Core purposes, strengths & use cases, especially re: Unity v2022 & beyond, mobile & AR/VR/XR platforms of today & tmrw..

3. OpenCV

- **Core Purpose**: A comprehensive computer vision library focused on image processing, video analysis, and classical computer vision algorithms
- **Key Strengths**:
  - Extensive set of traditional computer vision algorithms (feature detection, object tracking, camera calibration)
  - Mature C++ codebase with bindings for multiple languages including C#
  - High-performance image processing operations
  - Hardware acceleration support

- **Unity Integration**: Requires wrappers like OpenCVForUnity or native plugins

2. MediaPipe

- **Core Purpose**: Google's framework for building multimodal (video, audio, time-series) applied ML pipelines
- **Key Strengths**:
  - Pre-built solutions for hand tracking, face detection, pose estimation, and object detection
  - Graph-based architecture for ML pipeline construction
  - Optimized for mobile devices and real-time applications
  - Cross-platform with mobile-first approach
- **Unity Integration**: Available through Unity packages like MediaPipe Unity Plugin or custom native plugins https://github.com/homuler/MediaPipeUnityPlugin/releases/tag/v0.16.0

1. ONNX Runtime

- **Core Purpose**: High-performance inference engine for ONNX (Open Neural Network Exchange) models
- **Key Strengths**:
  - Model portability across frameworks (TensorFlow, PyTorch, etc.)
  - Optimized inference across CPU, GPU, and specialized hardware
  - Consistent performance across multiple platforms
  - Focused on efficient neural network execution
- **Unity Integration**: Works with Unity through Barracuda, Sentis, or custom native plugins

Key Differences

1. **Scope**:
   - OpenCV: Broad computer vision library with minimal ML focus
   - MediaPipe: End-to-end ML pipelines with pre-built solutions
   - ONNX Runtime: Purely an inference engine for neural network models
2. **Implementation Approach**:
   - OpenCV: Manual algorithm implementation and customization
   - MediaPipe: Graph-based pipelines with pre-trained models
   - ONNX Runtime: Model-centric approach with framework portability
3. **Mobile/XR Development**:
   - OpenCV: Requires careful optimization for mobile/XR
   - MediaPipe: Built specifically for mobile performance
   - ONNX Runtime: Optimized for cross-platform deployment
4. **Integration Complexity in Unity**:
   - OpenCV: Typically higher integration effort
   - MediaPipe: Moderate integration with built-in solutions
   - ONNX Runtime: Streamlined through Sentis for Unity's neural network inference

For Unity XR applications, these technologies are often used in combination - MediaPipe for real-time tracking, OpenCV for specialized vision processing, and ONNX Runtime for running custom ML models.

As of week of 3/3/25

Best Cross-Platform Options (Ranked by Speed & Performance)

1. Unity Sentis (Best Native Unity Solution)

- **Pros**: Native Unity solution, optimized for all platforms, compute shader acceleration, direct Unity integration
- **Cons**: Still somewhat new, limited model format support
- **Best For**: Simple to medium complexity models where Unity-native workflow is important
- **Example**: **GitHub - Unity Sentis Samples**

  BLOCKED: spent part of sprint 302 trying to develop a workaround. It may be possible with more custom work, but not worth it as Sentis works out of box with Unity v2023 / Unity 6

2. Barracuda/ONNX Runtime (Excellent Balance)

- **Pros**: Mature ONNX support, broad model compatibility, good optimization
- **Cons**: Not as deeply integrated with Unity, can be slower than platform-specific solutions
- **Example**: **GitHub - Unity Barracuda Samples**

3. Platform-Specific Native Solutions (Highest Performance)

For iOS/macOS:

- **CoreML + Metal**: Fastest possible iOS performance, Apple silicon optimized
- **Implementation**: **GitHub - CoreML-in-Unity**

https://github.com/apple/coremltools/tree/main

For Android:

- **TFLite + NNAPI**: Best performance on Android
- **Implementation**: **GitHub - TensorFlow-Lite-Unity**

4. OpenCV for Unity (Best for Computer Vision Algorithms)

- **Official Package**: **OpenCV for Unity**
- **Open Source Alternative**: **GitHub - EasyOpenCV**

Solution Architecture Recommendation

Per Claude 3.7 as of 3/9/25

For maximum speed and cross-platform compatibility, I recommend a hybrid approach:

1. **Unity Sentis Core Engine**: Use for cross-platform model execution and Unity-native workflow
2. **Platform-Specific Optimizations**:
   - For iOS: Implement critical computer vision operations in Metal/CoreML
   - For Android: Use TFLite/NNAPI for on-device acceleration
   - Wrap these in Unity native plugins
3. **Custom C++ Bridge** (for maximum performance):

```
cpp
CopyInsert
// iOS implementation
#if UNITY_IOS
extern "C" {
    bool InitializeNeuralEngine(const char* modelPath) {
        // CoreML setup
    }
}
#elif UNITY_ANDROID
extern "C" {
    bool InitializeNeuralEngine(const char* modelPath) {
        // TFLite + NNAPI setup
    }
}
#endif
```

High-Performance GitHub Examples

1. **High-Performance Face Detection**: **GitHub - MediaPipe Unity Plugin**
   - Extremely fast ML pipeline for iOS and Android
   - Face, hand, pose tracking in realtime
2. **OpenVINO-Unity**: **GitHub - OpenVINO for Unity**
   - Intel's high-performance inference engine
   - Great for x86 platforms (desktop)
3. **MNN Unity**: **GitHub - Alibaba MNN** **
   - Alibaba's blazing fast neural network inference
   - Optimized for mobile platforms
   - Lower level than other options but extremely fast

Mobile Optimization Tips

ONNX to ORT conversion

https://onnxruntime.ai/docs/execution-providers/CoreML-ExecutionProvider.htmln

On mobile, *.onnx model is not recommended, although it's supported. Best practice is to **Convert the Onnx model to Ort format.**

```
# Recommend using python virtual environment
pip install onnx
pip install onnxruntime
# In general,
# Use --optimization_style Runtime, when running on mobile GPU
# Use --optimization_style Fixed, when running on mobile CPU
python -m onnxruntime.tools.convert_onnx_models_to_ort your_onnx_file.onnx --optimization_style Runtime
```

Real-Time Object Detection in Unity w/ONNX Runtime &DirectML

Learn how to integrate a native plugin within the Unity game engine for real-time object detection using ONNX Runtime.

https://christianjmills.com/posts/unity-onnxruntime-cv-plugin-tutorial/#conclusion

> *" the main performance benefit of using ONNX Runtime over Unity's Barracuda or Sentis libraries is the specialized Execution Providers available for ONNX Runtime. The Execution Providers provide interfaces to hardware-specific libraries like Intel's OpenVINO and NVIDIA's TensorRT. These libraries allow you to get the most out of the hardware and support lower-precision inference. The drawback is that you need separate code for each hardware platform you want your Unity application to support. The main benefit of Unity's libraries is that they are cross-platform. "*

https://christianjmills.com/posts/unity-barracuda-inference-yolox-walkthrough/

https://github.com/cj-mills?tab=repositories

https://github.com/cj-mills/barracuda-inference-yolox-demo-urp

https://christianjmills.com/posts/pytorch-train-object-detector-yolox-tutorial/

https://yolox.readthedocs.io/en/latest/demo/onnx_readme.html#download-onnx-models

https://christianjmills.com/posts/unity-deep-learning-image-preprocessor-walkthrough/

https://christianjmills.com/posts/unity-yolox-utils-walkthrough/

https://christianjmills.com/index.html#category=unity

https://github.com/cj-mills/barracuda-inference-yolox-demo

as of 3/3

Sentis / ONNX

https://unity.com/products/sentis

https://www.youtube.com/watch?v=okSYmGrPBDE

Code

https://github.com/Unity-Technologies/sentis-samples

Models

https://huggingface.co/docs/hub/en/unity-sentis

https://huggingface.co/models?search=unity/sentis

https://github.com/onnx/models/tree/main?tab=readme-ov-file

MOBILE OPTIMIZATION

ONNX to ORT conversion

https://onnxruntime.ai/docs/execution-providers/CoreML-ExecutionProvider.htmln

On mobile, *.onnx model is not recommended, although it's supported. Best practice is to **Convert the Onnx model to Ort format.**

```
# Recommend using python virtual environment
pip install onnx
pip install onnxruntime
# In general,
# Use --optimization_style Runtime, when running on mobile GPU
# Use --optimization_style Fixed, when running on mobile CPU
python -m onnxruntime.tools.convert_onnx_models_to_ort your_onnx_file.onnx --optimization_style Runtime
```

COMPUTER VISION

ARFoundation (ARkit + ARcore)

Unity's framework for building AR experiences across multiple platforms.

- Supports: Device tracking, plane detection, image tracking, face tracking

https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/index.html (our version - Unity 2022 compatible)

Latest AR Foundation 6 Documentation:
https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.1/manual/index.html

GitHub Examples:

- Official ARFoundation Samples
- UnityARFoundationEssentials by Dilmer Valecillos

Unity Asset Store: AR Foundation Demo

https://assetstore.unity.com/packages/tools/utilities/ar-foundation-remote-2-0-201106

OpenCV for Unity

Unity plugin for the popular OpenCV computer vision library.

- Supports: Image processing, feature detection, object tracking

Documentation: OpenCV for Unity Documentation

GitHub Example: OpenCV for Unity Examples

Unity Asset Store:

- OpenCV for Unity
- OpenCV Plus Unity

YOLO (You Only Look Once)

Real-time object detection system, available through various Unity integrations.

- Supports: Fast object detection and recognition

Documentation: YOLO Documentation (Note: Unity integration may vary)

GitHub Examples:

- YOLOv8Unity
- Ultralytics YOLOv8
- YOLO-UnityBarracuda

Additional Resources:

- YOLO and Unity Perception Package Tutorial
- YOLOv8 Colab Tutorial
- Unity Forum Discussion on YOLO Integration
- Unity Barracuda Inference with YOLOX Walkthrough
- Roboflow Unity YOLO Model

Barracuda (replaced by Sentis)

Unity's lightweight cross-platform neural network inference library.

- Supports: Running neural networks for various CV tasks

Documentation: Barracuda Documentation

GitHub Examples:

- Official Barracuda Examples
- Unity Barracuda Inference Base

Unity Asset Store: MoodMe Emotions Barracuda SDK

Additional Resource: Unity Sentis (Beta) Discussion

ONNX Runtime

Cross-platform machine learning model inference engine, usable in Unity.

- Supports: Running pre-trained models for various CV tasks

Documentation: ONNX Runtime Documentation

GitHub Example: ONNX Runtime Unity Examples

MediaPipe

Google's framework for building multimodal ML pipelines, with Unity integration options.

- Supports: Face detection, hand tracking, pose estimation

Documentation: MediaPipe Documentation

GitHub Example: MediaPipe Unity Plugin

Additional Resource: MediaPipe Studio LLM Inference Demo

Other Advanced Techniques

Gaussian Splatting

Efficient 3D scene representation using oriented 3D Gaussians

GitHub Example: SplatVFX by Keijiro Takahashi

Related Application: Luma AI (iOS App)

Stable Diffusion in Unity

Integrating text-to-image generation capabilities in Unity

GitHub Examples:

- UnityMLStableDiffusion by Keijiro Takahashi
- ML Stable Diffusion by Keijiro Takahashi

AVATARS / AGENTS / NPCs

AI-Powered NPC Interactions

Unity Asset Store:

NPC AI Engine: Dialog, Actions, Voice and Lipsync (Convai)

AUDIO AI Libs

Whisper Speech-to-Text

See our Whisper repository in Plastic

Tested in iOS build & ready for integration

There may be other Sentis-based or alternative Whisper implementations as well, but the above was the fastest, most robust, local / device-based AI model available at time (no API calls)

Fish Audio: The Best & Free Generative AI Text To Speech & Voice Cloning

Powerful, fast, and customizable text-to-speech solution. Ultra-low latency, rapid voice cloning, and flat-rate pricing for AI Voice Over.

https://fish.audio/

Free Text to Speech & AI Voice Generator | ElevenLabs

Create the most realistic speech with our AI audio tools in 1000s of voices and 32 languages. Easy to use API's and SDK's. Scalable, secure, and customizable voice solutions tailored for enterprise needs. Pioneering research in Text to Speech and AI Voice Generation.

https://elevenlabs.io/

hart

🎙️ Audio ML Papers

🆕 https://github.com/camenduru/ACE-Step-jupyter

https://github.com/camenduru/dia-jupyter

https://github.com/camenduru/csm-1b-jupyter

https://github.com/camenduru/DiffRhythm-jupyter

https://github.com/camenduru/FluxMusic-jupyter

https://github.com/camenduru/FoleyCrafter-jupyter

https://github.com/camenduru/stable-audio-jupyter

https://github.com/camenduru/ChatMusician-jupyter

https://github.com/camenduru/NeMo-ASR-jupyter

https://github.com/camenduru/Image2SoundFX-jupyter

https://github.com/camenduru/metavoice-jupyter

https://github.com/camenduru/MAGNeT-colab

https://github.com/camenduru/resemble-enhance-colab

https://github.com/camenduru/OpenVoice-colab

https://github.com/camenduru/singing-voice-conversion-colab

https://github.com/camenduru/styletts-colab

https://github.com/camenduru/HierSpeech_TTS-colab

https://github.com/camenduru/AudioSep-colab

https://github.com/camenduru/coqui-XTTS-colab

https://github.com/camenduru/VALL-E-X-colab

https://github.com/camenduru/seamless-m4t-colab

https://github.com/camenduru/audiogen-colab

https://github.com/camenduru/LP-Music-Caps-colab

https://github.com/camenduru/vampnet-colab

https://github.com/camenduru/tortoise-tts-colab

https://github.com/camenduru/MusicGen-colab

https://github.com/camenduru/elevenlabs-colab

https://github.com/camenduru/Retrieval-based-Voice-Conversion-WebUI-colab

https://github.com/camenduru/whisper-jax-colab

https://github.com/camenduru/bark-colab

https://github.com/camenduru/audioldm-colab

MISC

Coding Resources

- Windsurf
- Claude Sonnet 3.7

Additional Considerations for XR Storytelling

- Real-time performance optimization
- Cross-platform compatibility
- Integration with Unity's rendering pipeline
- Balancing visual quality and computational resources
- User privacy and data handling
- Collaborative features for multi-user experiences
- Leveraging AI for dynamic content generation and NPC interactions