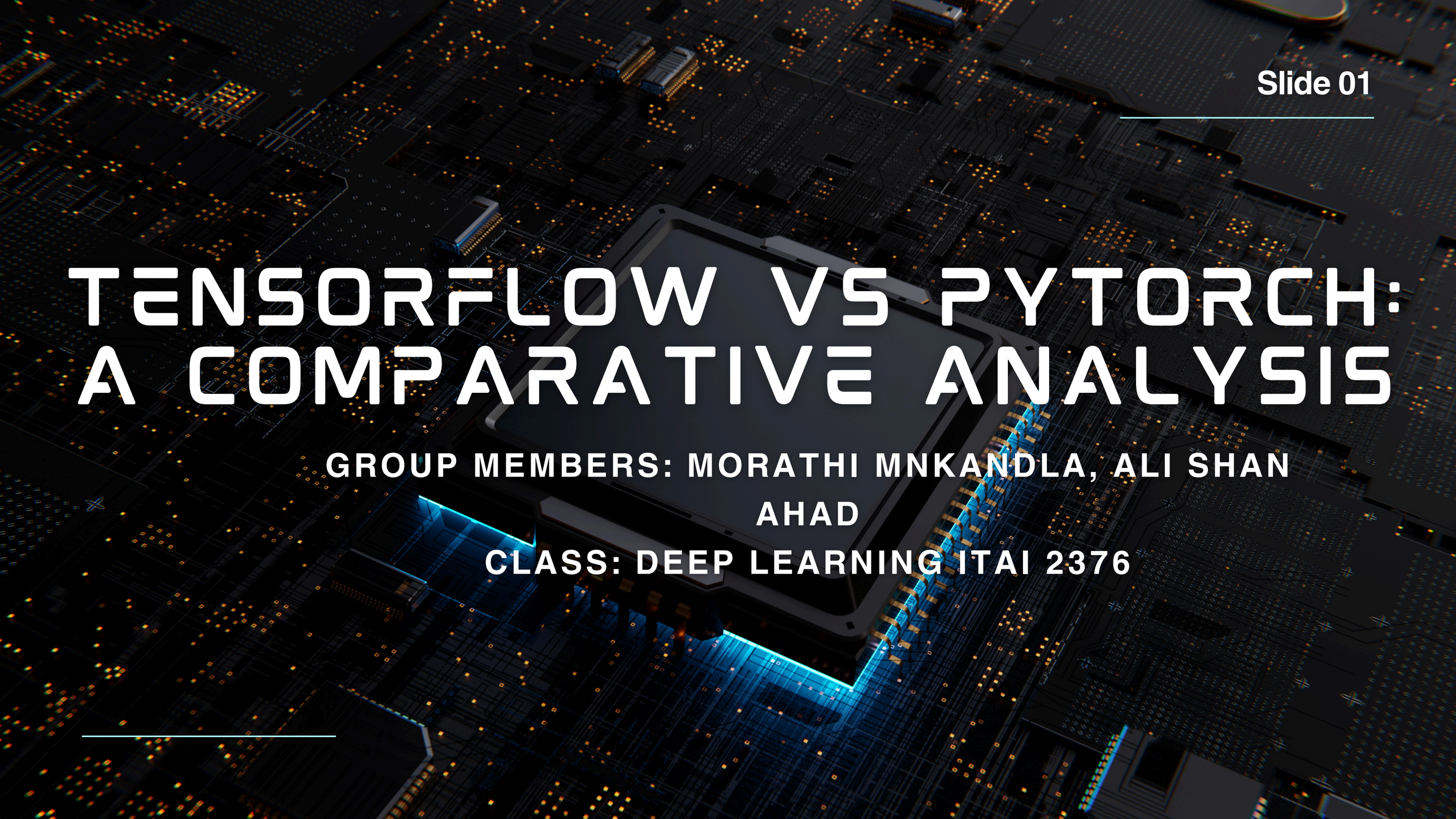


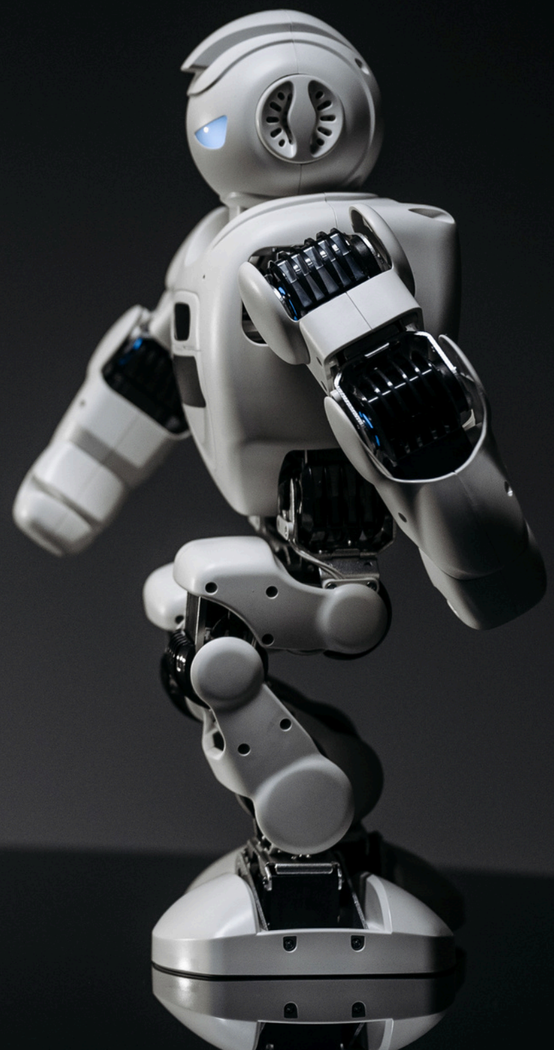
TENSORFLOW VS PYTORCH: A COMPARATIVE ANALYSIS

GROUP MEMBERS: MORATHI MNKANDLA, ALI SHAN AHAD

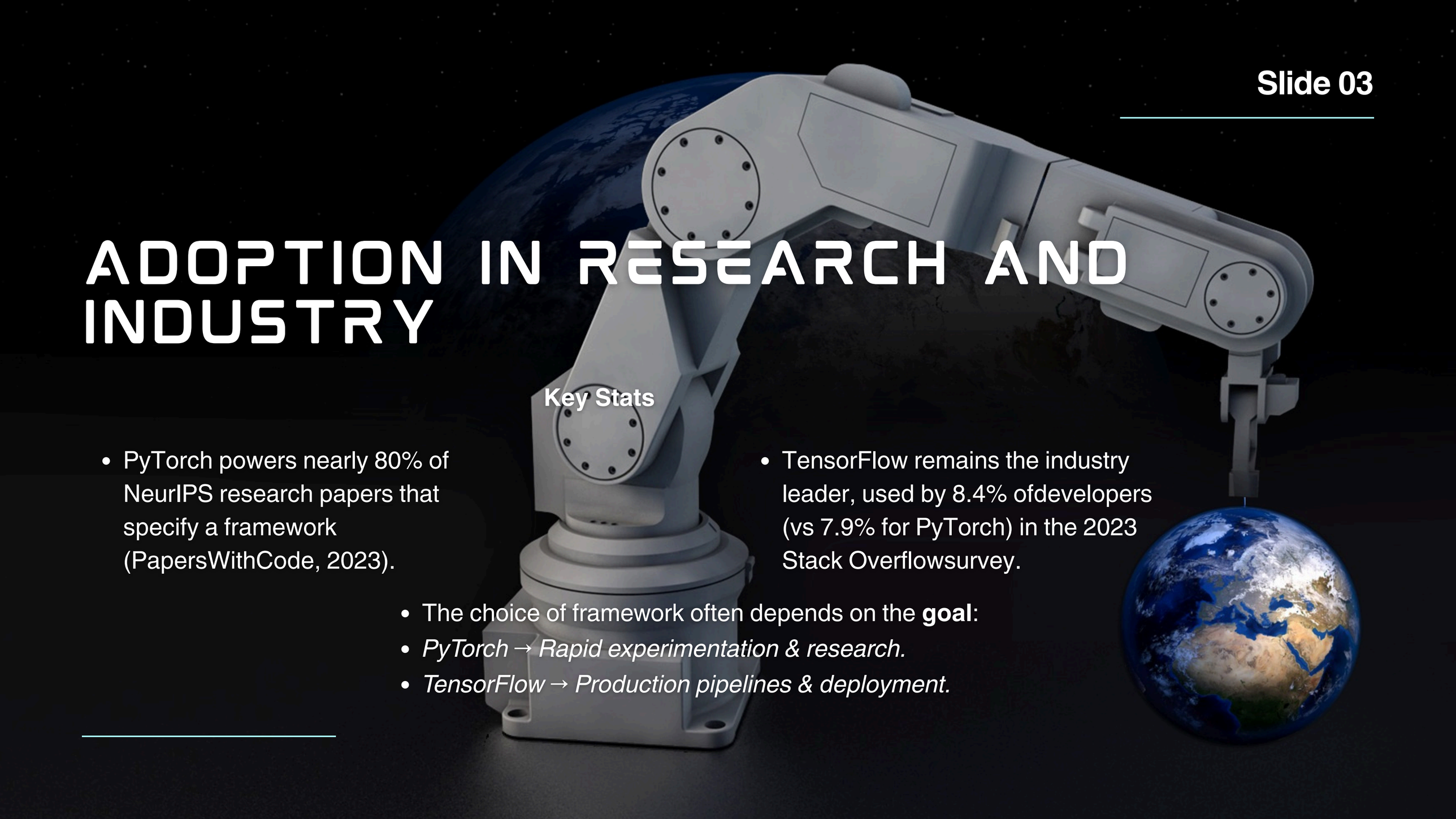
CLASS: DEEP LEARNING ITAI 2376

KEY POINTS:

- TensorFlow (2015) – Developed by Google Brain for scalable,production-ready machine learning.
- PyTorch (2016) – Created by Meta AI with a research-first, dynamic computation approach.
- These two frameworks dominate the AI landscape, driving innovation in both academia and industry.
- TensorFlow focuses on deployment and performance at scale, while PyTorch leads in research flexibility.



ADOPTION IN RESEARCH AND INDUSTRY



Key Stats

- PyTorch powers nearly 80% of NeurIPS research papers that specify a framework (PapersWithCode, 2023).
- TensorFlow remains the industry leader, used by 8.4% ofdevelopers (vs 7.9% for PyTorch) in the 2023 Stack Overflowsurvey.

- The choice of framework often depends on the **goal**:
- *PyTorch → Rapid experimentation & research.*
- *TensorFlow → Production pipelines & deployment.*

Slide 04

TENSORFLOW OVERVIEW

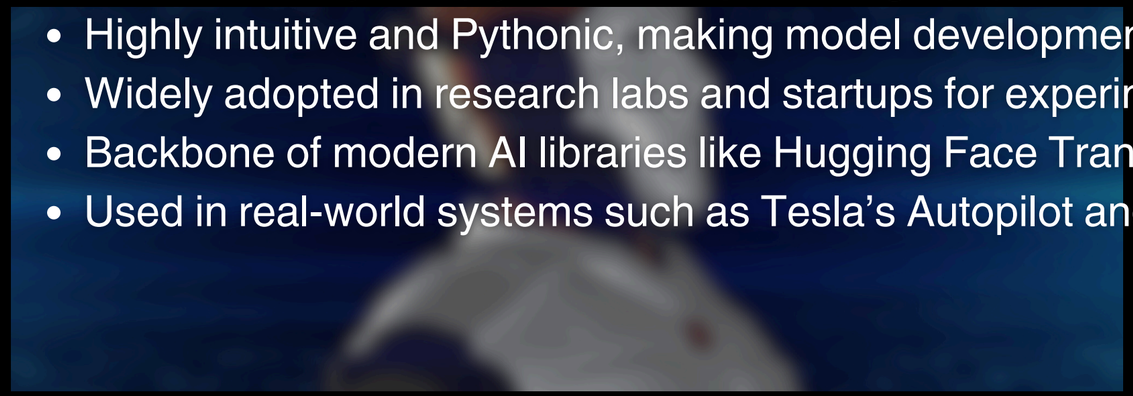
Points:

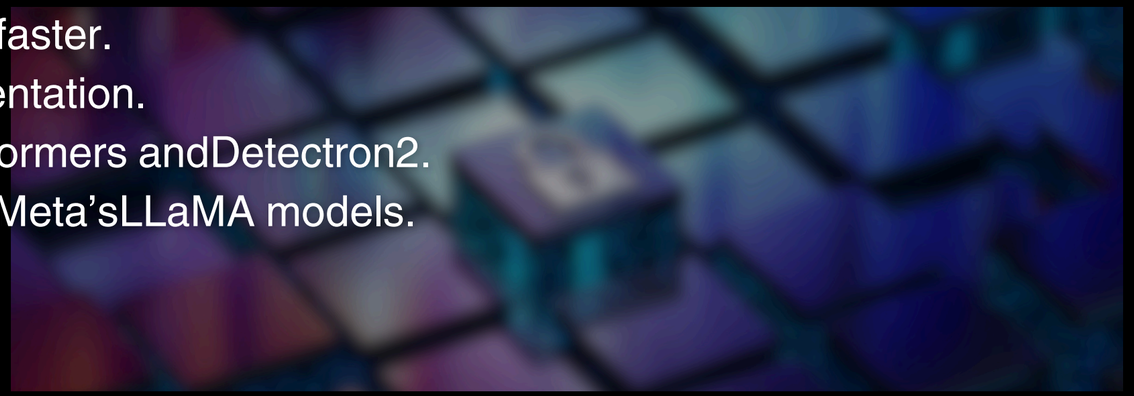
- Developed by Google Brain and released in 2015.
- Originated from Google’s internal DistBelief project.
- Initially used static computation graphs for performanceoptimization.
- TensorFlow 2.x (2019) introduced eager execution and integratedKeras for easier model building.
- Deployed across Google products: Search, Gmail, Translate,YouTube, Photos.
- Supported by a strong ecosystem including TensorFlow Lite,TensorFlow.js, and TF Serving.
-

PYTORCH OVERVIEW

Points:

- Released by Meta AI (Facebook) in 2016.
- Derived from the Torch framework, rewritten in Python.
- Uses define-by-run (dynamic computation graph) for real-timeexecution.
- Highly intuitive and Pythonic, making model development faster.
- Widely adopted in research labs and startups for experimentation.
- Backbone of modern AI libraries like Hugging Face Transformers andDetectron2.
- Used in real-world systems such as Tesla’s Autopilot and Meta’sLLaMA models.





TOTAL TRAINING TIME COMPARISON

Data Insight:

- PyTorch: 16.98 hours
- TensorFlow: 21.95 hours
- (Values from Novac et al., Sensors 2022.)

Interpretation:

- PyTorch completed training ~23% faster than TensorFlow on the same workload.
- Indicates stronger runtime efficiency and lower overhead for dynamic computation graphs.
- TensorFlow performs competitively but shows slightly higher training latency under similar settings.

Slide 07

EPOCH AND STEP TIMING

Data Insight:

- Average Epoch Time:
  - TensorFlow → 1.53 hours
  - PyTorch → 1.18 hours
- Per Training Step:
  - TensorFlow → 0.00112 s
  - PyTorch → 0.00087 s

Interpretation:

- PyTorch achieves faster per-step execution, especially on GPU-optimized workloads.
- TensorFlow’s improvements in version 2.x narrow the gap through XLA and graph optimization.

.

Slide 08

INFERENCE FOR SMALL IMAGE MODELS (28×28)

\Data Insight:

- TensorFlow (Keras): 0.8985 s/image
- PyTorch: 0.3032 s/image

Interpretation:

- PyTorch achieves ≈3× faster inference on lightweight models.
- Reflects lower overhead and efficient runtime execution for small data inputs.
- TensorFlow compensates with better integration for mobile and embedded deployment.

.

Slide 09

INFERENCE FOR LARGER MODELS & DEPTH NETWORKS

Data Insight:

- 64×64 images:
  - TensorFlow → 2.0182 s
  - PyTorch → 1.5961 s
- **Depth model test:**
  - TensorFlow → 2.667 s
  - PyTorch → 1.174 s

Interpretation:

- PyTorch consistently maintains lower inference latency for larger models.
- TensorFlow, however, supports hardware acceleration (TPU/XLA) that can offset raw speed differences.

.

OVERALL FEATURE COMPARISON (SCALE OF 0-10)

TensorFlow:

- Usability → 7
- Deployment → 9
- Performance → 8
- Community → 8

PyTorch:

- Usability → 9
- Deployment → 7
- Performance → 8
- Community → 9

Summary:

- PyTorch dominates in usability and research community engagement.
- TensorFlow excels in deployment and ecosystem integration.
- Performance is effectively equal when optimized properly.

DEPLOYMENT STRENGTHS (0 = WEAK, 3 = STRONG)

TensorFlow Strengths:

- Mobile / Edge — 3 (via TensorFlow Lite)
- Server — 3 (via TensorFlow Serving)
- Browser — 3 (TensorFlow.js)
- TPU Support — 3 (native integration)
- ONNX Interoperability — 2

PyTorch Strengths:

- Mobile / Edge — 1
- Server — 2
- Browser — 1
- TPU Support — 1 (via XLA backend)
- ONNX Interoperability — 3

Summary:

- TensorFlow dominates in deployment and portability.
- PyTorch leads in cross-framework flexibility with ONNX.

FINAL COMPARATIVE SUMMARY

Highlights:

- PyTorch is ideal for researchers and developers seeking agility and dynamic modeling.
- TensorFlow remains the best for scalable production systems and cross-platform deployment.
- Performance parity exists for well-optimized models.
- Both frameworks continue to evolve — their ecosystems now overlap through ONNX and shared APIs.
- Recommendation:
  - Prototype in PyTorch
  - Deploy in TensorFlow (or via ONNX pipelines)

REFERENCES & RESOURCE

Sources Cited:

- Ba Alawi — Comparative Survey of PyTorch vs TensorFlow (2025).
- Novac et al. (2022) — Sensors Journal: training and inferencebenchmarks.
- Becirovic et al. (2021) — comparative inference analysis for CNNmodels.
- PapersWithCode (2023) — framework usage in NeurIPS publications.
- Stack Overflow Developer Survey (2023) — AI framework adoptionrates.
- TensorFlow & PyTorch official documentation for feature verification.

THANK YOU

FOR YOUR ATTENTION