KI in Medienanwendungen

Seminar im Sommersemester 2026

Prof. Dr.-Ing. Uwe Hahne

02 Examples (as motivation)

The state of the art in Generative AI

- text generation ✅
- image generation ✅
- video generation (a few seconds, low resolution) ✅
- 3D model generation (low poly, simple textures) ✅
- audio generation (music, speech, sound effects) ✅
- code generation (Junior developer tasks) ✅

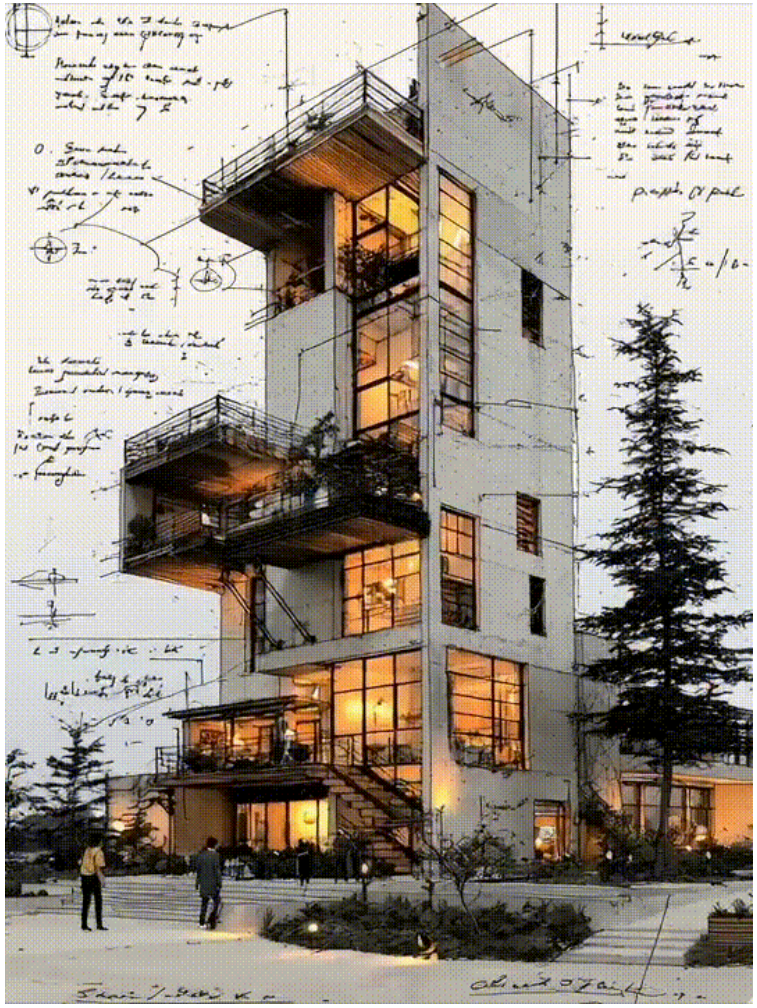
Leaderboards of Generative AI

- Check out Arena (formerly LMArena) for a comprehensive overview of the latest models and their capabilities: Arena
- Find a suitable model in the Hugging Face model hub: Hugging Face
- Another leaderboard focused on generatice AI models is the GenAI-Arena by TIGER lab, which is part of the University of Waterloo, Canada.

Further examples of generative AI applications

The next slides show some more examples that had been gathered in September 2025.

- Person image: This Person Does Not Exist

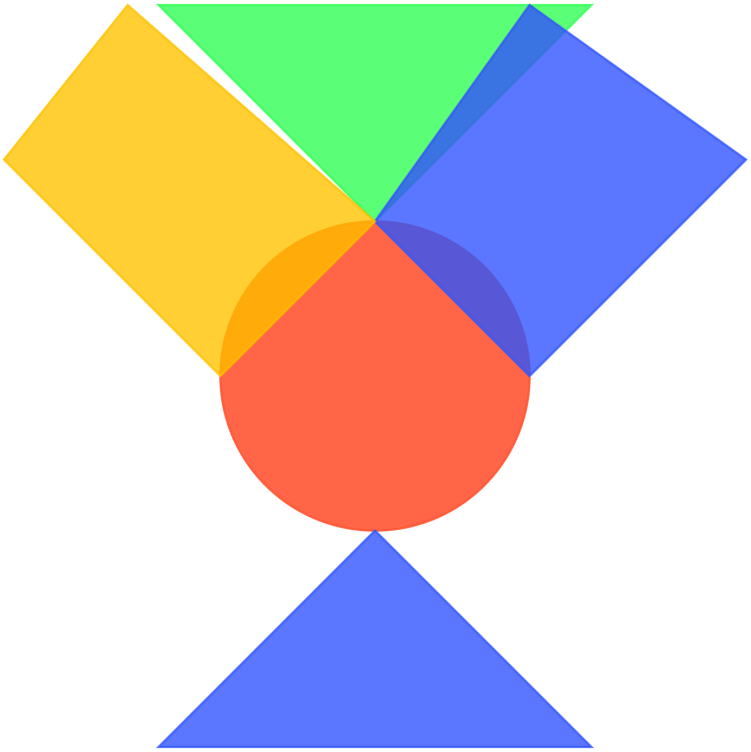


- Midjourney house draft animation: Midjourney Explore

- VEO3 racecar video: Google AI Studio

- Common Sense Machine texture: Blog with other examples, shown example was published only on Discord.

- Microsoft Research Trellis teaser: Trellis



- Logo generated by bwGPT: bwGPT using the code shown in the next slide.

```python
import matplotlib.pyplot as plt
import numpy as np

# Set up the figure
fig, ax = plt.subplots(figsize=(6, 6), facecolor='white')
ax.set_xlim(-1.5, 1.5)
ax.set_ylim(-1.5, 1.5)
ax.axis('off')

# Define colors (four colors)
colors = ['#FF5733', '#33FF57', '#3357FF', '#FFC300']

# Draw a central circle (representing creativity and the core of generative AI)
circle = plt.Circle((0, 0), 0.5, color=colors[0], alpha=0.9)
ax.add_artist(circle)

# Draw shapes around the circle (representing branching ideas or outputs)
# Top triangle
triangle_x = [0, 0.7, -0.7]
triangle_y = [0.5, 1.2, 1.2]
ax.fill(triangle_x, triangle_y, color=colors[1], alpha=0.8)

# Right diamond shape
diamond_x = [0.5, 1.2, 0.5, 0]
diamond_y = [0, 0.7, 1.2, 0.5]
ax.fill(diamond_x, diamond_y, color=colors[2], alpha=0.8)

# Left abstract shape
abstract_x = [-0.5, -1.2, -0.8, 0]
abstract_y = [0, 0.7, 1.2, 0.5]
ax.fill(abstract_x, abstract_y, color=colors[3], alpha=0.8)

# Bottom shape (inverted triangle)
bottom_triangle_x = [-0.7, 0, 0.7]
bottom_triangle_y = [-1.2, -0.5, -1.2]
ax.fill(bottom_triangle_x, bottom_triangle_y, color=colors[2], alpha=0.8)

# Save and display the logo
plt.tight_layout()
plt.show()
```