

Les 18 — Lesstof

Voice, vision, image gen, local LLMs, edge AI

Vak: AI-Assisted Development
Klas: Klas A (3 uur fysiek, laatste les!)
Vervolg op: Les 17 — Production Polish
Volgende: Eindopdracht (thuis)

Vak: AI-Assisted Development **Vorige les:** Les 17 — AI Production Polish **Volgende les:** Geen — eindopdracht thuis

Inhoud

1. [Het AI-landschap 2026](#1-het-ai-landschap-2026)
2. [Voice — Whisper + TTS](#2-voice--whisper--tts)
3. [Realtime API — voice chat met lage latency](#3-realtime-api--voice-chat-met-lage-latency)
4. [Vision — foto's analyseren](#4-vision--fotos-analyseren)
5. [Image generation — Flux + DALL-E](#5-image-generation--flux--dall-e)
6. [Local LLMs — Ollama](#6-local-llms--ollama)
7. [Edge AI — Vercel AI Gateway + Cloudflare Workers AI](#7-edge-ai--vercel-ai-gateway--cloudflare-workers-ai)
8. [Wanneer welke modaliteit](#8-wanneer-welke-modaliteit)
9. [Resources + community](#9-resources--community)

1. Het AI-landschap 2026

Wat is er beschikbaar

Models per provider (eind 2026):

Provider	Text	Vision	Voice	Image
OpenAI	GPT-5, o3	GPT-5	GPT-realtime, Whisper, DALL-E 4	DALL-E 4

Provider	Text	Vision	Voice	Image
Anthropic	Claude Opus 4.6, Sonnet 4.6	Claude 4.6	(via partners)	—
Google	Gemini 2.5 Pro/Flash	2.5 Pro	(via partners)	Imagen 3
Meta	Llama 4 (open)	Llama 4 Vision	—	—
xAI	Grok 4	Grok 4 Vision	—	Grok image
Black Forest	—	—	—	Flux 1.1 Pro
ElevenLabs	—	—	TTS (best in class)	—

Modi om te draaien

- **Cloud API** — direct via provider
- **Edge** — Vercel AI Gateway, Cloudflare Workers AI
- **Lokaal** — Ollama, llama.cpp, LM Studio

Vandaag zien we alle drie.

2. Voice — Whisper + TTS

Whisper voor transcriptie (audio → text)

```
import { experimental_transcribe as transcribe } from "ai";
import { openai } from "@ai-sdk/openai";

const result = await transcribe({
  model: openai.transcription("whisper-1"),
  audio: audioBlob, // Blob, Buffer, of file path
});
console.log(result.text);
```

Whisper varianten:

- whisper-1 — OpenAI cloud, 99 talen, \$0.006/minute
- whisper-large-v3 — open-source, lokaal te draaien
- Distil-Whisper — sneller, ~6× kleiner

TTS — text → audio

```
import { experimental_generateSpeech as speak } from "ai";

const result = await speak({
  model: openai.speech("tts-1-hd"),
  text: "Welkom bij Polderfest 2027!",
  voice: "alloy", // alloy, echo, fable, onyx, nova, shimmer
});

// result.audio.uint8Array of result.audio.base64
```

Voor productie: ElevenLabs voor natuurlijkere stemmen, of OpenAI tts-1-hd. Voor demo: tts-1 is goed genoeg.

Volledige voice-loop

```
// 1. User spreekt → audio blob
const transcription = await transcribe({ model, audio: audioBlob });

// 2. LLM antwoordt op tekst
const { text } = await generateText({
  model: openai("gpt-4o-mini"),
  prompt: transcription.text,
});

// 3. AI spreekt antwoord
const speech = await speak({ model: openai.speech("tts-1"), text });

// 4. Play in browser
const audio = new Audio(`data:audio/mp3;base64,${speech.audio.base64}`);
audio.play();
```

Latency: ~2-4 seconden total. Voor "echte" voice-chat met lagere latency: Realtime API.

3. Realtime API — voice chat met lage latency

Wat is anders

Standaard voice-loop heeft 3 calls: Whisper + LLM + TTS. Elk een roundtrip.

OpenAI Realtime API = één persistent WebSocket. Audio streamt naar OpenAI, audio streamt terug. Latency 200-500ms.

Features:

- Interruptions (user kan AI onderbreken)
- Function calling (tools tijdens voice)
- Verschillende voices
- Multi-modal (text + voice in zelfde sessie)

Setup (concept)

```
// Browser
const ws = new WebSocket("wss://api.openai.com/v1/realtime?model=gpt-realtime");

ws.onopen = () => {
  ws.send(JSON.stringify({
    type: "session.update",
    session: { voice: "alloy", instructions: "Je bent een helpdesk." }
  }));
};

ws.onmessage = (e) => {
  const event = JSON.parse(e.data);
  if (event.type === "response.audio.delta") {
    playAudio(event.delta); // streaming audio
  }
};

// User microphone audio → ws.send(audio)
```

Voor demo: complex maar magisch. Voor productie: hire een specialist of gebruik bestaande wrappers.

4. Vision — foto's analyseren

Basis call

```
const result = await generateText({
  model: openai("gpt-4o"),
  messages: [{
    role: "user",
    content: [
      { type: "text", text: "Beschrijf wat er op deze foto staat." },
      { type: "image", image: "https://example.com/foto.jpg" },
      // of: { type: "image", image: buffer }, voor lokale file
    ],
  }],
});
```

Multiple images

```
content: [
  { type: "text", text: "Vergelijk deze twee foto's." },
  { type: "image", image: url1 },
  { type: "image", image: url2 },
],
```

Use cases

- **OCR** — tekst uit foto's halen
- **Defect detection** — productie-foto's checken
- **Visual search** — "toon producten zoals deze"
- **Accessibility** — automatische alt-tekst
- **Receipt scanning** — bonnetjes naar gestructureerde data
- **Code from screenshot** — Excel-screenshot → JSON

Cost

Vision is 2-5× duurder dan tekst voor zelfde input. Reden: image wordt naar veel tokens omgezet (een 1024×1024 plaatje = ~1100 tokens).

Provider keuze

- GPT-4o — beste algemeen, snel
- Claude Sonnet 4.6 — beste voor diagrammen en tabellen
- Gemini 2.5 — goed + cheap

5. Image generation — Flux + DALL-E

Flux via Fal

```
import { experimental_generateImage as generateImage } from "ai";
import { fal } from "@ai-sdk/fal";

const { image } = await generateImage({
  model: fal.image("fal-ai/flux/dev"),
  prompt: "Een retro festival-poster voor Polderfest 2027, oranje en cream kleuren, vinyl",
  size: "1024x1024",
});

// image.base64 of image.url
```

Flux varianten:

- `flux/schnell` — snelle, goedkope variant (~\$0.003/image)
- `flux/dev` — middle ground
- `flux/pro` — hoogste kwaliteit (~\$0.05/image)

DALL-E via OpenAI

```
const { image } = await generateImage({
  model: openai.image("dall-e-3"),
  prompt: "...",
  size: "1024x1024",
  quality: "hd",
});
```

DALL-E is duurder maar bekender. Voor productie: Flux meestal de winnaar (kwaliteit + cost).

Goede prompts

Slecht: "Een mooie afbeelding"

Goed: "Een retro festival-poster, stijl van 70s rockconcert-flyers, hoofdtitel 'POLDERFEST 2027' in art-deco lettertype, oranje en cream kleurpallet, decoratieve geometrische randen, geen mensen, op afgeleefd papier"

Specifieker = beter. Style, palette, compositie, wat NIET te tonen.

Use cases

- Landing page hero-images
- Social media posts on-the-fly
- Avatar-generatie per user
- Product-mockups
- Email illustrations

Storage

Images zijn groot. Voor productie:

- Generate → upload to Supabase Storage of S3
- Save URL in database
- Gebruik CDN voor delivery

6. Local LLMs — Ollama

Wat is Ollama

Tool om open-source LLMs lokaal te runnen. Eén command per model. Werkt op macOS, Linux, Windows.

Setup

```
# Install
brew install ollama # macOS
# of via download van ollama.com

# Pull a model
ollama pull llama4:8b
ollama pull qwen3:14b
ollama pull mistral-small:24b
```

Run

```
ollama run llama4:8b
# Interactieve chat in terminal — 100% lokaal
```

Via AI SDK

```
pnpm add ollama-ai-provider
```

```
import { createOllama } from "ollama-ai-provider";

const ollama = createOllama({
  baseUrl: "http://localhost:11434/api",
});

const result = await generateText({
  model: ollama("llama4:8b"),
  prompt: "Wat zijn de hoofdfuncties van een AI Developer?",
});
```

Wanneer lokaal

Voordelen:

- Privacy — data verlaat je laptop niet
- Cost — geen API-cost
- Offline — werkt zonder internet
- Latency — vaak snel op M-series Mac

Nadelen:

- Kwaliteit < GPT-5 / Claude voor complex
- Hardware vereist (8GB+ RAM voor klein, 32GB+ voor groot)
- Eerste keer downloaden duurt lang (5-50GB per model)

Use cases:

- POC zonder cost

- Privacy-critical apps (healthcare, legal)
- Offline tools (developer tooling)
- Cost-sensitive massa-apps

Aanrader modellen 2026

Model	Size	Best voor
llama4:8b	5GB	Algemeen, klein
qwen3:14b	9GB	Code, reasoning
mistral-small:24b	15GB	Best kwaliteit / size
llama4:70b	40GB	Top kwaliteit

7. Edge AI — Vercel AI Gateway + Cloudflare Workers AI

Vercel AI Gateway

Eén endpoint voor alle providers. Voordelen:

- **Unified API** — switch providers met 1 line change
- **Fallback chains** — als OpenAI down, automatisch Claude
- **Cost optimization** — cheapest available model
- **Key management** — Vercel beheert keys, jij niet
- **Caching** — built-in response caching

```
import { vercel } from "@ai-sdk/vercel";

// Direct gebruik
const result = await generateText({
  model: vercel("openai/gpt-4o"),
  prompt: "...",
});

// Met fallback
const result = await generateText({
  model: vercel.fallback([
    "openai/gpt-4o",
    "anthropic/claude-sonnet-4.5",
    "google/gemini-2.5-flash",
  ]),
  prompt: "...",
});
```


Cloudflare Workers AI

LLMs op Cloudflare's edge (250+ datacenters wereldwijd). Lage latency overall.

```
pnpm add @ai-sdk/cloudflare
```

```
import { workersai } from "@ai-sdk/cloudflare";

const result = await generateText({
  model: workersai("@cf/meta/llama-3.1-8b-instruct"),
  prompt: "...",
});
```

Models beschikbaar:

- Llama 4 (klein + groot)
- Mistral, Qwen
- Whisper (voice)
- Stable Diffusion (image gen)

Cost: gratis tier 10k requests/dag. Daarna betaal per gebruik.

Wanneer edge

- Globale app, latency belangrijk
- Cost-sensitive (free tiers ruim)
- Geen vendor lock-in
- Vercel deploy stack (logical extension)

8. Wanneer welke modaliteit

Vuistregel — denk per feature:

Feature	Beste modaliteit
FAQ-zoeken	Text RAG
Foto-upload met analyse	Vision
Handsfree input	Voice
Marketing-asset generen	Image gen
Real-time conversation	Voice (Realtime API)
Privacy-critical data	Local LLM
Globale snelle response	Edge AI

Feature	Beste modaliteit
Massa goedkope calls	Edge AI of Local

Niet één tool voor alles — combineer.

9. Resources + community

Communities

- **r/LocalLLaMA** — Reddit, dagelijkse model-updates
- **AI Engineer Foundation** — Discord, productie-focus
- **Latent Space** — podcast + Discord, breed
- **Hugging Face Discord** — modellen + papers
- **AI SDK Discord** — Vercel's eigen

Blogs + nieuws

- **Simon Willison** — <https://simonwillison.net> (#1 AI blog)
- **Latent Space** — wekelijks podcast
- **The Rundown AI** — daily newsletter
- **Stratechery** — strategisch perspectief (paid)

Twitter/X-accounts

- @karpathy, @swyx, @sama, @AnthropicAI, @OpenAIDevs, @simonw

Voor diepere kennis

- **Karpathy Zero to Hero** — neural networks vanaf nul (YouTube, gratis)
- **Andrew Ng Courses** — Coursera, structureel
- **Hugging Face NLP course** — gratis, hands-on

Conferenties

- **AI Engineer Summit** — SF, jaarlijks
- **NeurIPS, ICLR** — research-focus
- **Devbase, MAINSTAGE** — Nederlands

Bronnen — alle docs

- **AI SDK voice + vision:** <https://ai-sdk.dev/docs/ai-sdk-core/transcription>
- **AI SDK image gen:** <https://ai-sdk.dev/docs/ai-sdk-core/image-generation>

-
- **Ollama models library:** <https://ollama.com/library>
 - **Vercel AI Gateway:** <https://vercel.com/docs/ai/ai-gateway>
 - **Cloudflare Workers AI:** <https://developers.cloudflare.com/workers-ai>
 - **Fal models:** <https://fal.ai/models>
 - **Replicate models:** <https://replicate.com>
 - **OpenAI Realtime:** <https://platform.openai.com/docs/guides/realtime>
 - **ElevenLabs:** <https://elevenlabs.io>