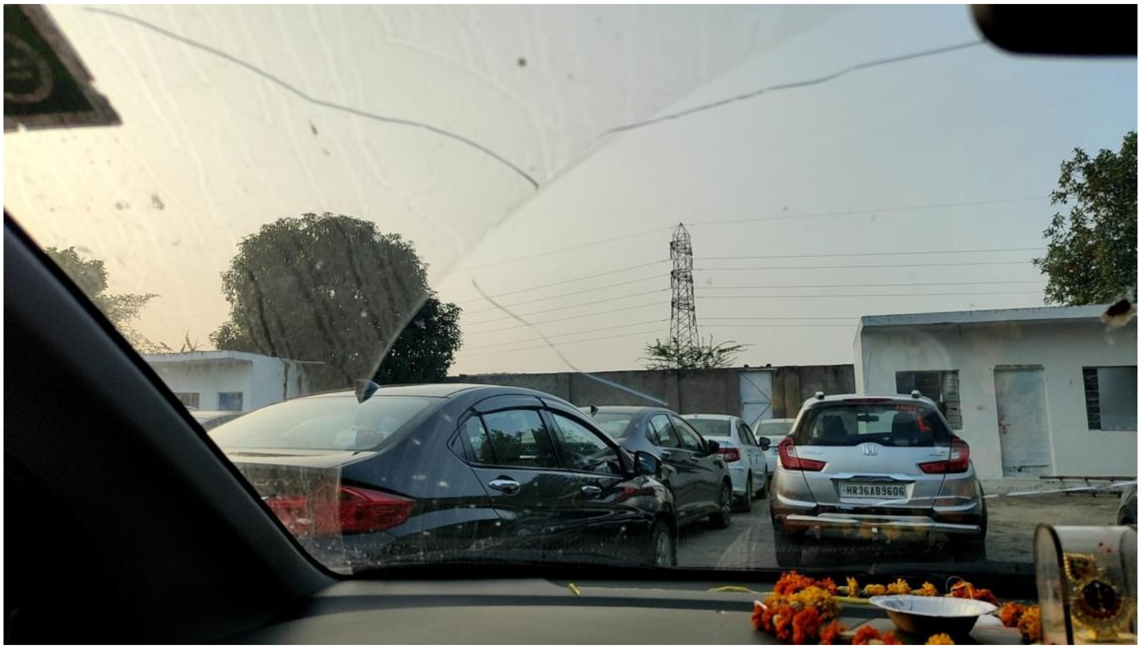
Enhancing Windshield Damage Assessment with integration of Claim Genius's Proprietary GenAI with Claude’s Model Context Protocol (MCP)



Is This Windshield Damaged?

Enhancing Windshield Damage Assessment with Claim Genius's Proprietary GenAI damage detection and Anthropic Claude’s Model Context Protocol (MCP) Framework. Default LLMs like Claude or ChatGPT fail to provide domain specific solution to complex problems like Windshield Damage Detection but integration with Claim Genius’s GenAI MCP tool server allows state of the art LLMs to make correct judgement on windshield damages in seconds.

Executive Summary

Identifying windshield damage from vehicle photos is inherently challenging due to the **complex visual environment** in which images are captured. Several external factors can distort the image and lead to **false positives or misclassification**, including:

- **Lighting Conditions**: Overexposure, reflections, or insufficient lighting can obscure or exaggerate the appearance of cracks or chips.

- **Glare and Reflections**: Sunlight or reflections from surrounding surfaces can mimic the appearance of damage.
- **Shadows from Trees, Buildings, or Overhead Wires**: These can create visual artifacts that resemble cracks or scratches on the windshield.
- **Image Angle and Quality**: The orientation and resolution of the image can affect the visibility of actual damage and make detection more difficult.

This variability makes it extremely difficult for **generic AI models** to reliably assess windshield condition. They often lack the **contextual awareness** to distinguish between actual damage and environmental noise.

At **Claim Genius**, our **proprietary GenAI pipeline** is designed to overcome these limitations. By combining **specialized models** (for detecting image angle, part presence, and environmental cues) with a **domain-trained LLM**, we ensure that the final judgment is made based on contextual intelligence, not just raw pixels.

In the automotive insurance industry, accurate and timely damage assessment is critical - especially for high-frequency claim areas such as windshield damage. While general-purpose large language models (LLMs) like Claude and ChatGPT offer general insights, they lack the domain-specific intelligence needed for visual and contextual understanding.

This white paper presents **Claim Genius's proprietary GenAI and ML-based architecture**, enhanced by a **MCP system**, that dramatically improves the accuracy, speed, and reliability of windshield damage assessments. By orchestrating a network of specialized AI tools and leveraging a domain-trained LLM, our system turns complex image data into actionable claim-level insights.

The Problem: Why General-Purpose LLMs Fall Short

Insurance adjusters, like John in our case study, increasingly rely on AI tools to manage their growing workload. When John adopted Claude Desktop, he gained basic support with text-based tasks - but complex visual challenges, like determining windshield damage, remained out of reach.

Generic LLMs are not designed for:

- Interpreting visual automotive data
- Detecting damage location and severity in images
- Understanding spatial context or part orientation
- Coordinating multiple sources of structured visual data

Asking a general-purpose LLM to perform specialized damage assessment is like asking a general physician to perform heart surgery - the tool is simply not trained for the task.

The Claim Genius Solution: Specialized Intelligence with LLM + MCP

At Claim Genius, we took a different approach. We designed an GenAI system that mirrors how a trained human adjuster works, breaking down the task into focused subcomponents, each handled by a specialized tool.

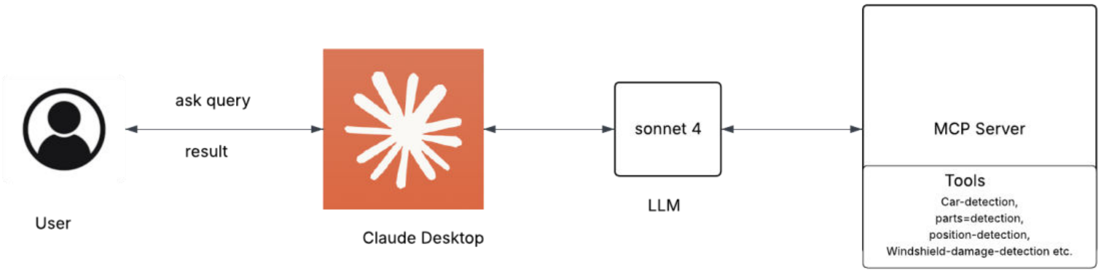
At the core of this system is the MCP framework - a lightweight AI orchestration server that manages multiple purpose-built models. These models analyze specific aspects of the image and pass structured outputs to our Claim Genius proprietary LLM, which synthesizes the results into a comprehensive, claim-level decision.

MCP Pipeline Components

Each AI model in the pipeline solves a dedicated sub-task:
- Car Detection: Verifies the presence of a vehicle in the image.
- Part Detection: Identifies visible components, such as windshield, bumper, and door.
- Position Detection: Determines the viewing angle or image orientation of the car.
- Windshield Damage Detection: Analyzes the windshield for chips, cracks, and breakage.
- Claim-Level Aggregation: Synthesizes outputs across multiple images into one claim decision.

All these tools are hosted as independent API services and coordinated by the MCP server. The structured output is then passed to Claude Desktop, powered by Claim Genius's LLM, which performs final reasoning and generates human-readable insights.

System Architecture



Client Input: Multiple vehicle images per claim
MCP Server: Coordinates tool execution
AI Toolchain: Each tool processes images independently

Proprietary LLM: Synthesizes tool outputs, reasons over structured data
Final Output: Accurate, contextual claim-level damage summary

This architecture mirrors real-world thinking - instead of guessing from raw pixels, the system compiles expert-level findings into a unified report.

Real-World Impact: John's Story

Before Claim Genius:
- John manually reviewed 15-20 images per claim.
- Windshield damage assessments took hours.
- The process was stressful and error-prone.

After Claim Genius:

- The system flagged only relevant images.
- Damage assessments took seconds.
- Confidence scores and summaries made verification easy.

John now spends more time resolving claims and less time combing through images - thanks to the power of

Focused Claim Genius AI.

Why Claim Genius's Approach Works

By integrating multiple narrow, high-accuracy models and combining their output using a domain-trained reasoning engine, our system delivers:

- Higher Precision: Specialized tools reduce false positives and missed detections.
- Greater Contextual Awareness: Position and part recognition provide accurate spatial reasoning.
- Fewer Hallucinations: LLM only reasons over trusted, structured outputs.
- True Claim-Level Intelligence: Goes beyond individual images to understand the full case.

Implementation Details

Environment Setup

- MCP server is installed via server.py
- Integrated with Claude Desktop via config file
- Tools are exposed as callable APIs (e.g., /car-detection, /windshield-damage)

Tech Stack

- Python, FastMCP
- httpx for API calls
- Claude Desktop for reasoning and presentation

- Install claude desktop - https://claude.ai/download
- Create a mcp_server - server.py and register your Claude Genius MCP Server with Anthropic Claude.
- Add mcp_server into claude_desktop_config.json -

```
{
  "mcpServers": {
    "claim_server": {
      "command": "uv",
      "args": [
        "--directory",
        "E:\\mcp server",
        "run",
        "server.py"
      ]
    }
  }
}
```

Install Packages to run MCP server

```
uv pip install httpx "mcp[cli]"
```

Sample Code

server.py

```python
from mcp.server.fastmcp import FastMCP
import signal
import sys
import httpx


# Handle SIGINT (Ctrl+C) gracefully
def signal_handler(sig, frame):
    print("Shutting down server gracefully...")
    sys.exit(0)


signal.signal(signal.SIGINT, signal_handler)


# Create an MCP server with increased timeout
mcp = FastMCP(
    name="claim_server",
    host="127.0.0.1",
    port=5000,
    timeout=30,  # Increase timeout to 30 seconds,
)
```

```python
@mcp.tool(description="Detects whether in given image car is present.")
def car_detection_tool(image_url: str)->dict:
    """
    This tool is used to detect whether in a given image a car is present or not.


    Args:
    image_url: url of the image to be processed.


    Returns:
    dict: A dictionary containing a car present or not with confidence.
    """
    data = {
        "data": {
            "images": [image_url]
        }
    }

    try:
        response = httpx.post(
            "https://example.api.car-detection-tool/",
            json=data,
        timeout=10
        )
        response.raise_for_status()
        output = response.json()
        return output['output']
    except Exception as e:
        return {"error": str(e)}


@mcp.tool(description="identify angle of car .")
def position_detection_tool(image_url: str) -> dict:
    ...


@mcp.tool(description="detect car parts present in image.")
def parts_detection_tool(image_url: str) -> dict:
    ...


@mcp.tool(description="detect windshield is damaged or not.")
def windshield_damage_detection_tool(image_url: str) -> dict:
    ...
```

```
@mcp.tool(description="aggrate image level results into claim level")

def claim_level_aggregation_tool(image_urls: list[str]) -> dict:

    ...

if __name__ == "__main__":

    mcp.run()
```

Conclusion: GenAI Built for Insurance

Claim Genius's proprietary GenAI architecture, powered by Claude's MCP framework, offers a powerful leap beyond generic AI solutions. By combining focused perception models with high-level language reasoning, we empower adjusters to process windshield damage claims with unmatched speed and precision.

This is not just automation - it's domain-specific intelligence designed for real-world insurance workflows.

About Claim Genius

Claim Genius is a global leader in AI-powered claims automation for the automotive industry. With cutting-edge image intelligence and LLM innovation, we're enabling insurers to process claims faster, reduce fraud, and improve customer experience across markets.