Fix Three Critical Bugs in DMS Pipeline

The following three issues are currently disrupting the Document Management System (DMS) pipeline. Please review the associated files and implement the requested fixes to resolve them.

Issue 1: Redundant Files in Pipeline (Fix in dms_scan.py)

Problem Description:
The dms scan operation, and subsequent pipeline steps, are resulting in file duplication. Specifically, after dms_image_to_text.py runs, both the original image file (e.g., Doc/IMG_4666.jpeg) and the generated text file (Doc/md_outputs/IMG_4666.jpeg.txt) are being included in the pending list for summarization. This is redundant, as the text file contains the content intended for summarization.

Required Fix in dms_scan.py:
When constructing the new_files list, a check must be implemented. If a file (the original image) has a corresponding processed text file in the md_outputs/ directory, the **original file must be excluded** from the new_files list. The expectation is that only the generated text file (if new) will proceed for summarization.
The logic should be: *For every image file found, check if its .txt twin exists in md_outputs/. If the twin exists, skip the image file.*

Issue 2: Disruptive Preview in dms_review.py

Problem Description:
When running dms_review.py, the show_file_preview function attempts to display the raw content of image-to-text output files (e.g., .jpg.txt, .png.txt). This content is often raw, unformatted OCR output which causes visual disruption ("havoc") on the console screen and is not useful to the user during the review phase.

Required Fix in dms_review.py:
Modify the show_file_preview function to skip the file content preview under two conditions:

1. If the file has a **common image extension** (e.g., .png, .jpg, .jpeg, .gif), skip the preview.
2. If the file is a **generated text output** from image processing, which can be identified by checking if the string "md_outputs" is present in its path.

The function should only attempt to read and display content for genuine text/code/markdown files.

Issue 3: re.PatternError in dms_apply.py (Critical)

Problem Description:
The dms_apply.py script fails critically when attempting to apply approved changes due to a regular expression error during state block replacement:

```
re.PatternError: bad escape \u at position 375 (line 9, column 44)
```

This error occurs at the line: content = state_pattern.sub(new_state_block, content) in the update_dms_state function. The cause is that **backslashes (\)** within the new_state_block string (which contains JSON-formatted file paths and summaries) are being misinterpreted by Python's re.sub function as escape sequences (e.g., \n, \t, or the problematic \u).

Required Fix in dms_apply.py:

The safest way to inject an arbitrary string (like new_state_block) as a replacement pattern into re.sub is to use a **lambda function** as the replacement argument.

Modify the update_dms_state function to safely inject new_state_block using a lambda:

Change the problematic line from:

```
content = state_pattern.sub(new_state_block, content)
```

to:

```
content = state_pattern.sub(lambda m: new_state_block, content)
```

This ensures the new_state_block is treated as a literal replacement string, resolving the re.PatternError.